MSX DOS

A.C.J. Groeneveld

handboek voor iedereen



uw **MSX** *computer de baas*

MSX

DOS *handboek voor iedereen*

A.C.J. Groeneveld

MSX DOS
handboek voor iedereen

MSX DOS

handboek voor iedereen

uw **MSX** *computer*
de baas

A.C.J. Groeneveld

uitgeverij STARK - TEXEL
postbus 302 - 1794 ZG Oosterend tel. 02223 - 661

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Groeneveld, A.C.J.

MSX DOS handboek voor iedereen: uw MSX computer de baas/
A.C.J. Groeneveld.–Oosterend: Stark-Texel
ISBN 90 6398 674 2
SISO 365.3 UDC 681.3.06
Trefw.: programmeren (computer) / MSX DOS (computer).

....

1e druk 1985
ISBN 90 6398 407 3

INHOUD

In het grote MSX-basic handboek werd de MSX computer en het MSX-basic uitgebreid behandeld. De bezitter van een eenvoudige MSX computer vindt in dit grote handboek dan ook alle geheimen van de MSX computer uitgebreid beschreven.

In het MSX DISK handboek werd de MSX computer met schijveneenheid besproken. Het uitgebreide MSX basic, het MSX DISK basic, biedt de bezitter van een MSX systeem met schijveneenheid alle benodigde ondersteuning.

Dit MSX DOS handboek gaat in op het eerste professionele operating system dat speciaal voor hobby-computers werd ontworpen; het MSX DOS. Niet alleen het MSX DOS maar ook het verschijnsel operating system in het algemeen wordt in dit boek uitgebreid behandeld.

Met nadruk werd in dit boek rekening gehouden met het feit dat veel amateurs met het MSX DOS voor het eerst met een operating system te maken krijgen. Een uitgebreide inleiding tot het begrip operating system zorgt ervoor dat ook deze amateurs niet halverwege 'uit de rails vliegen'.

Alhoewel dit MSX DOS handboek een op zichzelf staand werk is, is het toch verstandig om voorafgaand aan bestudering van dit boek eerst uitgebreid te hebben geëxperimenteerd met een MSX computer met schijveneenheid; het liefst met behulp van het grote handboek en het MSX DISK handboek. Indien u nog niet in het bezit bent van (één van) deze boeken, dan kunt u deze bestellen bij uw computerboekhandel. Uw handelaar heeft aan het ISBN-nummer van het boek genoeg om het voor u te kunnen bestellen:

MSX BASIC handboek voor iedereen	407 blz	ISBN 90 6398 100 7
MSX DISK handboek voor iedereen	121 blz	ISBN 90 6398 407 3

Eventueel kunt u uw bestelling ook direkt bij de uitgever plaatsen (telefoon 02223 - 661).

Ik hoop dat dit 'MSX DOS handboek voor iedereen' de andere MSX handboeken van Stark-Texel in hun enorme succes mag volgen.

juni 1985
A.C.J. Groeneveld.

2.1 Permanente gegevensopslag

Het is bij een computer onontbeerlijk dat gegevens voor langere tijd kunnen worden bewaard. Een groot en ingewikkeld programma willen we niet elke avond opnieuw intikken, maar slechts éénmaal invoeren en vervolgens vastleggen. Standaard kan dit vastleggen van gegevens in MSX-basic gebeuren op een cassetteband met behulp van een cassette-recorder.

Elke MSX-programmeur stuit al snel tegen twee grote bezwaren van deze opslagmethode, namelijk:

– De opslag duurt vrij lang. 32 kilobytes (32768 tekens) op cassetteband vastleggen duurt ongeveer vijf minuten. Dit lijkt in eerste instantie snel, maar blijkt al vlug een groot bezwaar te zijn.

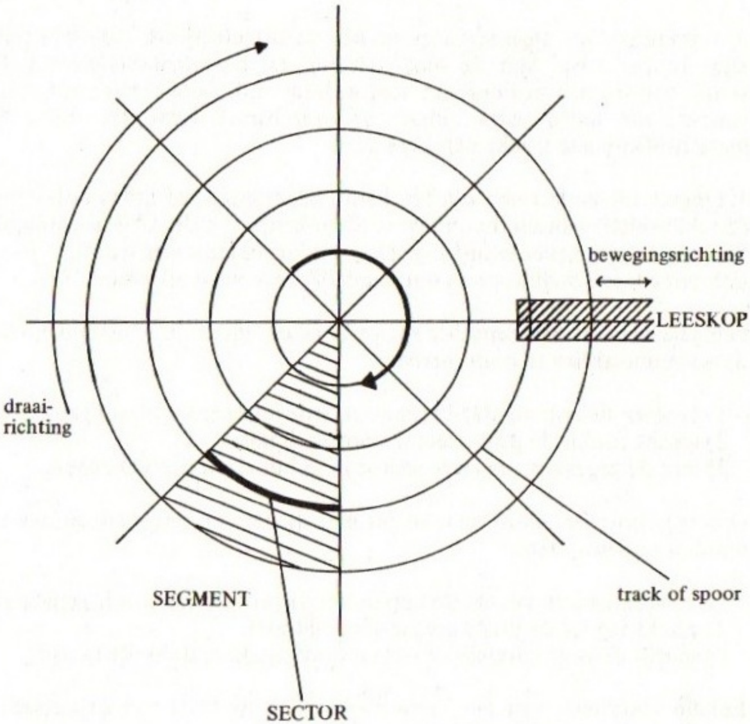
– De opslag is niet erg betrouwbaar. Vooral wanneer de cassetterecorder niet in topconditie is of verkeerd bandmateriaal wordt gebruikt, blijken gegevens tijdens opslag te kunnen worden verminkt. Het is dan ook noodzakelijk om uitstekend bandmateriaal aan te schaffen, de cassetterecorder regelmatig schoon te maken en af te stellen en de vastgelegde gegevens altijd te controleren.

Ondanks deze grote bezwaren is de cassetteband het meest voor de hand liggende opslagmedium voor hobby-computers; een cassetterecorder is relatief erg goedkoop, terwijl ook het bandmateriaal relatief zeer goedkoop is.

Wanneer snelheid bij het vastleggen en ophalen van gegevens een rol gaat spelen, wordt de cassetterecorder al snel een onbruikbaar randapparaat. Er dient dan naar een sneller opslagmedium te worden uitgekeken: de magneetschijf.

2.2 De magneetschijf

De magneetschijfeenheid (disk unit) gaat uit van het principe dat gegevens worden gelezen van en geschreven naar een snel ronddraaiende schijf. Deze schijf, van aluminium of kunststof, is bedekt met een magnetisch geprepareerde laag, die vergelijkbaar is met de laag magnetisch materiaal die op een cassetteband is aangebracht.



We kunnen een magneetschijf ingedeeld zien in een aantal 'taartpunten' die we segmenten noemen. Daarbij kunnen we ons allemaal om elkaar heen liggende cirkels op het schijfoppervlak voorstellen, de sporen of tracks. De doorsnede van een segment met een track noemt men een sector. We kunnen een magneetschijf dus onderverdeeld zien in een groot aantal sectoren, verdeeld over de verschillende tracks op het schijfoppervlak.

Een sector is de eenheid van uitwisseling met een schijf. Op een sector kan een blok gegevens, meestal ter grootte van 512 of 1024 bytes, in één keer worden geschreven. Ook wordt een blok altijd in één keer ingelezen.

Om gegevens uit een bepaalde sector te kunnen lezen of om een bepaalde sector te kunnen (her) schrijven, is een lees-schrijfkop nodig, net zoals

een opname- en weergavekop in een cassetterecorder noodzakelijk zijn. In het geval van de magneetschijf zit de gekombineerde leesschrijfkop op een beweegbare arm gemonteerd. Deze arm wordt door een speciale motor naar binnen of naar buiten bewogen, zodat de leesschrijfkop alle tracks kan bereiken.

Magneetschijven kunnen aan één kant (single sided) of aan twee kanten (double sided) worden beschreven, afhankelijk van de schijveneenheid. Wanneer een magneetschijf aan twee zijden beschreven wordt, zijn er ook twee lees/schrijfkoppen noodzakelijk; één voor elke kant.

Om gegevens uit een bepaalde sector te lezen, dient de schijveneenheid de volgende akties te ondernemen:

1) beweeg de arm zo, dat de kop boven de juiste track wordt geplaatst.
2) wacht totdat de juiste sector voorbij draait.
3) lees de gegevens van deze sector in in het computergeheugen.

Om gegevens op schijf vast te leggen, dienen de volgende akties te worden ondernomen:

1) beweeg de arm zo, dat de kop boven de juiste track wordt geplaatst.
2) wacht totdat de juiste sector voorbijdraait.
3) schrijf de gegevens vanuit het computergeheugen in die sector.

Enkele voordelen van de magneetschijf ten opzichte van de cassetterecorder vallen onmiddellijk op:

- Er behoeft geen cassetterecorder meer op opnemen of afspelen te worden gezet; de schijveneenheid zorgt zelf voor in- en uitschakeling voor 'opnemen' en 'afspelen' van gegevens.

- Alle sectoren zijn in onderdelen van seconden te bereiken; lange door-of terugspoeltijden komen hierdoor te vervallen.

- De schijveneenheid is speciaal voor gegevensopslag ontworpen. In tegenstelling tot een gewone cassetterecorder behoeft de schijveneenheid niet geschikt te zijn om muziek te registreren. Hierdoor kan de kwaliteit van een disk unit worden toegespitst op de opslag van gegevens hetgeen een sneller en veel betrouwbaarder opslaan van gegevens mogelijk maakt.

2.3 Vormen van magneetschijven

De magneetschijfeenheid of disk unit komt in diverse verschillende vormen voor. Drie vormen van magneetschijfeenheden behandelen we hieronder:

– de 3½ inch floppy disk unit

  Deze schijveneenheid maakt gebruik van 3,5 inch floppy disks. Deze floppy disks hebben een doorsnede van 3,5 inch (ongeveer 9 cm) en zijn verpakt in een hard plastic omhulsel. Het leesvenster is meestal afgedekt met een klepje en schuift bij het insteken van de floppy in de schijveneenheid automatisch open, waarna de lees/schrijfkop toegang heeft tot de daadwerkelijke schijf. De schijf zelf is gemaakt van een flexibele kunststof met daarop een magnetisch geprepareerde laag.
  Over het algemeen kent deze schijf-vorm een redelijk hoge opslag; per floppy disk kan meestal een hoeveelheid van 250.000 tot 500.000 tekens worden opgeslagen...

– de 5¼ inch floppy disk unit

  Deze schijveneenheid maakt gebruik van grotere floppy disks (5½ inch = ongeveer 13 cm doorsnede). Deze floppy disks hebben een open leesvenster en moeten daarom altijd in een speciale envelop worden bewaard. De verdere specificaties zijn ongeveer gelijk aan die van de 3½ inch floppy disk met dit verschil dat de opslagcapaciteit meestal wat hoger ligt (tot maximaal ongeveer 1.000.000 tekens).

– de winchester unit (hard disk)

  De schijf in een winchester unit kan niet worden uitgenomen. Meestal is een winchester disk gemaakt van een hard materiaal (aluminium) waaraan dan ook de benaming 'hard disk' is ontleend. Doordat de schijf vast in de schijveneenheid is opgenomen en daarbij van niet flexibel materiaal is gemaakt, is het voor de fabrikant mogelijk om de schijf harder te laten draaien en nauwkeuriger af te stellen. Grotere benaderingssnelheden en grotere opslagmogelijkheden zijn hiervan het gevolg. De gemiddelde winchester is al gauw weer 50 maal zo snel als een floppy en kan meestal 5 tot 10 miljoen tekens bergen...
  Door de hoge prijsstelling is een harde schijf meestal niet voor de hobbyist weggelegd.

2.4 Write protect

Bij elke vorm van magneetschijven is de mogelijkheid aanwezig om de gehele schijf te beschermen tegen abusievelijk overschrijven van gegevens. Deze mogelijkheid noemt men meestal de WRITE PROTECT optie.

De 3½ diskette bezit hiertoe in één van de hoeken een klein schuifje. Door dit schuifje zo te zetten dat een gaatje ontstaat, wordt het schijfje beschermd tegen het per ongeluk overschrijven van gegevens. Nadat het schuifje op beschreven wijze in de write protect stand is geschoven, is het voor de computer onmogelijk om nog gegevens op de schijf te schrijven. Het blijft echter mogelijk om gegevens van deze schijf te LEZEN.

De 5¼ inch floppy disk heeft geen schuifje. In plaats hiervan dient een plakkertje te worden geplakt over de kleine insparing die zich aan de kant van het leesvenster bevindt. Nadat dit plakkertje is aangebracht, is het voor de computer niet meer mogelijk om nog op de schijf te schrijven. LEZEN gaat natuurlijk nog wel.

De hard disk unit bezit meestal een write protect knopje waarmee de schijf onbeschrijfbaar kan worden gemaakt.

Bij optredende fouten bij het schrijven geldt altijd dat u eerst dient te controleren of de schijf niet via de WRITE PROTECT optie is beschermd.

Wanneer dit niet het geval blijkt te zijn, onderzoek dan of de schijf wel geformatteerd is (zie de volgende paragraaf).

2.5 Formatteren

We zagen reeds dat een magneetschijf is onderverdeeld in zogenaamde sectoren. Elke sector staat op een vaste plaats op de magneetschijf. In elke sector kan een blok gegevens (meestal 512 of 1024 bytes) worden geschreven.

Voordat een magneetschijf (bijvoorbeeld een floppy disk) in gebruik wordt genomen, dient de onderverdeling in sectoren eerst te worden aangebracht op de schijf. De magneetschijf moet als het ware voor het gebruik eerst in stukjes worden verdeeld.

Het verdelen van de magneetschijf in partjes noemt men het FORMATTEREN van de magneetschijf. In de meeste gevallen geldt:

EEN SCHIJF MOET EERST WORDEN GEFORMATTEERD, EERDER KAN HIJ NIET WORDEN GEBRUIKT.

Hoe een magneetschijf kan worden geformatteerd, wordt in hoofdstuk 7 behandeld.

2.6 De logische indeling van een magneetschijf

In de vorige paragrafen stelden we ondermeer vast dat we de magneetschijf uiteindelijk kunnen zien ingedeeld in een aantal sectoren. Elke sector bevat een blok van gegevens ter grootte van 512 of 1024 bytes. Zo'n blok is de kleinste eenheid die in zijn geheel kan worden gelezen of geschreven.

De indeling van de schijf in sporen, segmenten en sectoren (blokken) noemen we de fysische indeling van de schijf. Deze indeling kan als het ware op de schijf zelf worden aangewezen.

Met de logische indeling van de schijf bedoelen we de indeling die het MSX DOS hanteert, gebruik makende van de fysische indeling.

Doordat MSX DOS een intelligente en doordachte logische indeling van de magneetschijf ondersteunt, behoeft de gebruiker van de MSX-computer nooit stil te staan bij de fysische indeling van de magneetschijf. MSX bepaalt de fysische indeling, zorgt dat een blok niet ten onrechte door een ander kan worden overschreven en houdt bij in welke blokken welke gegevens zijn geschreven en welke blokken bij elkaar horen. Nooit behoeft de gebruiker dus een lijstje bij te houden van welke sectoren er werden gebruikt voor opslag van een programma. Door alleen maar de naam van het programma aan het MSX-systeem te verschaffen, kan worden bewerkstelligd dat de sectoren waarin het programma is opgenomen in de juiste volgorde worden ingelezen in het computergeheugen.

MSX DOS zorgt ervoor dat op elke disk een *inhoudsopgave* (index) wordt bijgehouden. In deze inhoudsopgave staat vermeld:

- wat de naam is van de verzameling van gegevens die op schijf werd geschreven.
- wat de soort is van deze verzameling van gegevens. Zijn het pro-

grammaregels of zijn het andere gegevens?

- welke sectoren worden bezet door deze gegevens. Totdat deze gegevens worden verwijderd, mogen de sectoren, waarin deze gegevens staan, niet voor andere doeleinden worden gebruikt. MSX DOS waakt hierover.

Een bij elkaar behorende verzameling van gegevens noemt men in de computerwereld over het algemeen een BESTAND of een FILE. De MSX-schijveneenheid is een bestandgeoriënteerd medium. Voordat gegevens op floppy worden geschreven, dient eerst een bestand te zijn gedefinieerd op schijf.

MSX DOS verzorgt de besturing van de bestanden (het file management) waardoor we als MSX DOS-gebruiker niet met de fysieke opbouw van de schijveneenheid worden gekonfronteerd.

3 BESTANDEN

Zoals in het vorige hoofdstuk werd opgemerkt, gaat het MSX DOS uit van een bestandsstruktuur op schijf. De term FILE of BESTAND is daarom erg belangrijk. Onthoudt:

EEN BESTAND OF FILE IS EEN GROEP VAN BIJ ELKAAR BEHORENDE GEGEVENS.

3.1 Bestandssoorten

MSX DOS onderscheidt naar funktie verschillende soorten bestanden, namelijk:

1. COMMAND FILES. Deze bestanden onderscheiden zich door het achtervoegsel .COM aan de naam van het bestand (zie hoofdstuk 8). Een commando file bevat een machinetaalprogramma en wordt door MSX DOS als kommando beschouwd. De gegevens uit een command file vormen bij elkaar dus een programma dat door MSX DOS direkt kan worden uitgevoerd zonder dat eerst het BASIC behoeft te worden geladen. Een command file is vaak het resultaat van een compilatie van bijvoorbeeld een PASCAL programma of van een assembleer-run. De professionele gebruikerssoftware die onder MSX DOS wordt aangeboden (databases, tekstverwerkers, compilers, enz.). wordt meestal in de vorm van één of meer command files aangeboden.

2. BATCH FILES. Deze bestanden onderscheiden zich door het achtervoegsel .BAT aan de naam van het bestand (zie hoofdstuk 8). Een batch file bevat een hoeveelheid kommando's zoals deze in hoofdstuk 9 worden behandeld en die direkt door het MSX DOS worden 'begrepen'. In een batch file kunnen opeenvolgingen van MSX DOS-kommando's die erg vaak voorkomen, worden gegroepeerd onder één naam. Zo kunnen onder MSX DOS door de gebruiker zelf nieuwe, complexe kommando's worden samengesteld. Een uitgebreide behandeling van de batch file volgt in hoofdstuk 10.

3. SYSTEEM FILES. Deze bestanden onderscheiden zich door het achtervoegsel .SYS aan de naam van het bestand (zie hoofdstuk 8). Een systeem file bevat gegevens die van direkt belang zijn voor het MSX DOS zelf; de gebruiker krijgt met dit soort bestanden

meestal niet te maken.

4. Overige bestandssoorten. MSX DOS maakt geen onderscheid tussen de overige nog mogelijke bestandssoorten. Het MSX DISK basic doet dit wel: met name maakt het MSX DISK basic een duidelijk onderscheid tussen RANDOM files en SEQUENTIAL files. Ook onderscheidt MSX DISK basic duidelijk de PROGRAM file van de DATA file. Het MSX DOS doet dit niet; alle overige bestandssoorten worden op een zelfde wijze behandeld.

3.2 Bestandsgrootten

De gebruiker van het MSX DOS zal op geen enkel gebied de noodzaak tegenkomen, de GROOTTE van een bestand te vermelden. Dit komt omdat het MSX DOS een zogenaamde dynamische behandeling van bestanden kent. Een bestand wordt altijd zo groot toegewezen als nodig is en kan later altijd worden aangevuld. De enige grens aan de grootte van een bestand wordt uiteindelijk gevormd door de opslagcapaciteit van de magneetschijf zelf.

4 OPERATING SYSTEMS

Het MSX DOS is een operating system voor de MSX computer. De gebruiker die dit leest, vraagt zich natuurlijk onmiddellijk af wat nu eigenlijk een operating system is. In dit hoofdstuk wordt uit de doeken gedaan wat de algemene principes van een operating system zijn en wat de achtergrond van het verschijnsel operating system is.

4.1 Operating systems overal

Iedereen die wel eens met een computer te maken heeft gehad, hoe klein ook, heeft al eens te maken gehad met een operating system. Met een beetje fantasie zou men zelfs kunnen beweren dat de eenvoudigste rekenmachine al een operating system heeft.

Wat een operating system nu daadwerkelijk is, wordt al wat duidelijk wanneer we de minder gebruikelijke nederlandse vertaling van het woord operating system nader beschouwen:

OPERATING SYSTEM = BESTURINGSSYSTEEM

De volgende definitie van een operating system is misschien niet helemaal correct maar in ieder geval wel duidelijk:

Een operating system (besturingssysteem) is een stuk programmatuur dat gedurende de werking van een computer altijd (gedeeltelijk) aanwezig is in het computergeheugen en dat de coördinatie van de verschillende computeronderdelen op het allerlaagste niveau verzorgt.

Wanneer we een eenvoudige MSX computer zonder schijveneenheid beschouwen, kunnen we wat betreft de aanwezige programmatuur in het ROM geheugen (Read Only Memory = permanent, alleen uitleesbaar geheugen) het volgende schema tekenen:

Wanneer u auto rijdt, kunt u zichzelf vergelijken met de MSX basic-interpretator. Het intelligente werk van koppelen, gas geven, sturen, remmen, uitwijken en dergelijke wordt door u gedaan. Uw auto vormt in deze vergelijking het kernel. De akties die u onderneemt, worden door de auto uiteindelijk effektief gemaakt. U draait aan het stuur; de auto verandert van richting. U haalt een handle over; het elektrische systeem van uw auto zorgt ervoor dat er richting wordt aangegeven.

Samenvattend kunnen we stellen dat het kernel de mogelijkheden biedt om de computer te kunnen besturen. De software van het tweede niveau bestuurt de computer daadwerkelijk door van de kernel gebruik te maken.

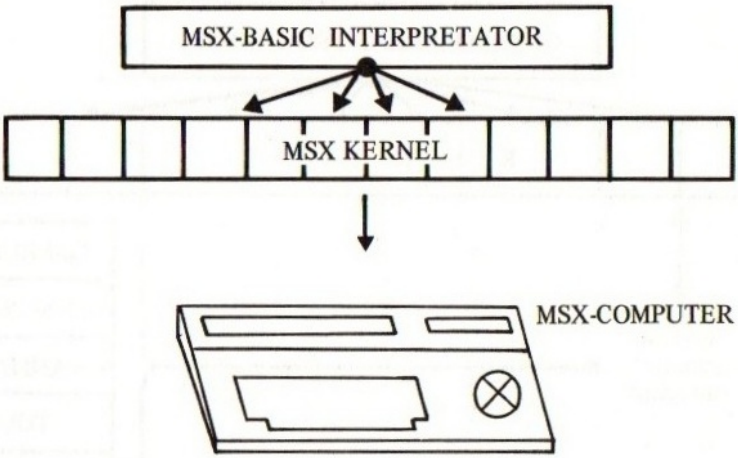
Het woord kernel is algemeen gebruikelijk als naamgeving van de verzameling van besturingsprogrammatuur van een computer. Een nederlands woord hiervoor is bijvoorbeeld KERNPROGRAMMATUUR. Ook het woord NUCLEUS is vrij gebruikelijk.

Deze besturingsmogelijkheid biedende programmatuur wordt vaak al aangeduid als BESTURINGSSYSTEEM of OPERATING SYSTEM. Als zodanig kan worden gesteld dat elke computer, hoe klein ook, een operating system heeft. Daar dit operating system meestal nogal verweven is met de basic-interpreter, wordt een dergelijk operating system ook wel een intrinsiek operating system genoemd. In de volgende paragraaf zullen we zien, dat er eigenlijk nog een element nodig is om daadwerkelijk van een operating system te kunnen spreken.

4.2 Een echt besturingssysteem

In de vorige paragraaf zagen we, dat in feite in elke computersysteem een operating system aanwezig is. Dit operating system bestaat uit een verzameling van besturende machinetaalprogramma's zonder dat er sprake is van een daadwerkelijke samenhang tussen deze programma's. De basic-interpreter die zich meestal op het tweede, coördinerende niveau bevindt, zorgt voor een samenhang van deze stukjes systeem en maken het operating system zinvol.

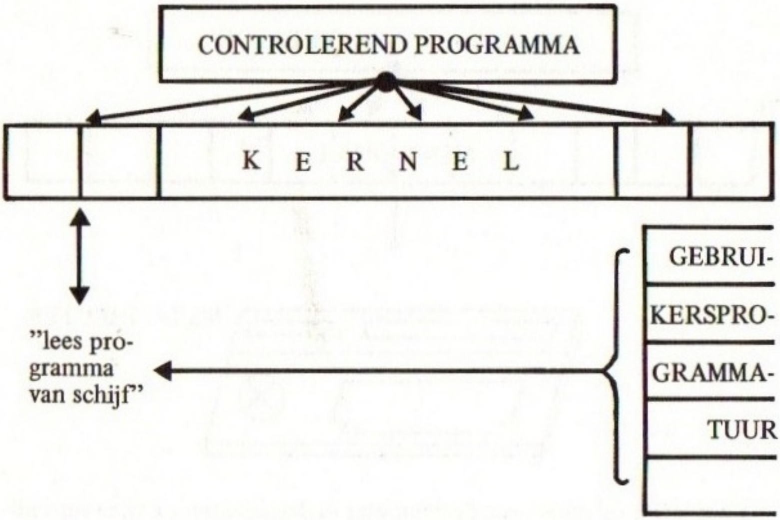
Een 'echt' operating system kan men zich schematisch als volgt voorstellen:



In het ROM geheugen van de computer onderscheiden we twee verschillende niveaux van (machinetaal-) software:

1. de MSX BASIC INTERPRETATOR. Deze software controleert de door u ingetoetste BASIC-kommando's op juistheid. Goed bevonden kommando's worden bij het uitvoeren van een door u samengesteld MSX basic programma door de interpretator daadwerkelijk uitgevoerd. De interpretator INTERPRETEERT in feite het door u ingetoetste BASIC programma en vertaalt 'uw woorden' naar aanwijzingen die door de computer kunnen worden begrepen.

2. het MSX KERNEL. Het MSX kernel wordt gevormd door een verzameling van tientallen meestal vrij kleine programma's. Deze programma's dragen elk zorg voor de daadwerkelijke besturing van een klein onderdeel van het totale computersysteem. Zo is er één programma dat de afhandeling met cassetterecorder verzorgt. Ook is er een programma dat de ingave via het toetsenbord mogelijk maakt. Weer een andere routine uit het MSX kernel draagt zorg voor het trekken van een lijn over het beeldscherm. Bij elkaar vormt het kernel de totale besturing van de MSX computer. De MSX basic interpretator die 'daar boven' staat, gebruikt deze routines uit het kernel en geeft deze de nodige gegevens mee. Het 'intelligente' werk wordt eerst door de MSX basic interpretator uitgevoerd waarna de elementaire akties door het kernel worden uitgevoerd.



De taakstelling van een operating system kan men als volgt zien:

- een operating system biedt een NUCLEUS, een verzameling van programmatuur waarmee de onderdelen van de computer kunnen worden bestuurd.

- een operating system biedt de mogelijkheid om elementaire akties met betrekking tot het computersysteem te ondernemen. Deze elementaire akties worden meestal betiteld als het ONDERHOUD van het systeem.

- een operating system biedt de mogelijkheid om controle met behoud van de NUCLEUS over te dragen aan een gebruikersprogramma.

Het controlerende programma, dat zelf ook uitgebreid gebruik maakt van de nucleus, biedt de mogelijkheid om het computersysteem te onderhouden. Zo kunnen bestanden met dit controlerende programma worden ingericht, verwijderd, gekopiëerd en kunnen bijvoorbeeld ook schijven worden geformatteerd. Ook, en dit is één van de belangrijkste mogelijkheden van het controlerende programma, heeft het de mogelijkheid om zichzelf 'op te offeren' ten gunste van een ander programma. Via de betreffende besturing in het kernel kan een ander

gebruikersprogramma (een basic interpreter, een pascal compiler, een tekstverwerker enzovoorts) in de plaats van het controlerende programma worden gezet terwijl de besturende software, de kernel, aanwezig blijft.

Het operating system cijfert zichzelf dus gedeeltelijk weg, maar laat een gedeelte van zichzelf (de kernel) staan voor gebruik door het vervangende programma. Wanneer dit vervangende programma klaar is met de uit te voeren taken, kan het controlerende programma met behulp van een routine uit de kernel weer worden geplaatst.

4.3 DOS

Alhoewel in elke hobbycomputer wel een operating system is aan te wijzen, komt het verschijnsel operating system als een op zichzelf staande grootheid uit de professionele computerwereld. Immers, een groot en duur computersysteem 'hangt' men in deze wereld niet aan één computertaal op. Het enige dat een professioneel computersysteem moet bieden, is een uitstekend operating system. Van dit operating system kunnen de software fabrikanten dan gebruik maken voor het ontwerpen van de benodigde computertalen en andere gebruikerssoftware.

Een professioneel computersysteem zonder schijveneenheid is, behoudens enkele specifieke besturingstoepassingen, nauwelijks voor te stellen. Het spreekt dan ook voor zich dat een operating system meestal voorziet in de mogelijkheid tot besturing van één of meer schijveneenheden. Sterker nog, de meeste operating systems zijn volledig gebaseerd op het gebruik van een schijveneenheid.

Wanneer, en dit is in de meeste gevallen zo, een operating system is gebaseerd op het gebruik van minstens één schijveneenheid, noemt men een dergelijk operating system meestal een Disk Operating System, afgekort DOS.

Het MSX DOS is een speciaal voor de MSX computer door Microsoft ontwikkeld disk operating system.

In het vorige hoofdstuk zagen we, wat een Disk Operating System (DOS) is.

Het MSX DOS is een disk operating system, speciaal voor MSX-computers ontworpen door Microsoft, een software-fabrikant van wereldfaam op de wat professionelere microcomputer.

Een goede software-producent vindt niet steeds opnieuw het wiel uit. Steeds maakt hij gebruik van eerdere produkten en ervaringen om de volgende, nieuwere produkten beter, foutlozer, sneller, logischer en doordachter te maken.

Op deze wijze stelde Microsoft behalve het MSX basic en het MSX DISK basic ook het MSX DOS operating system samen, gebruik makend van professioneel software-materiaal en jarenlange ervaringen op professioneel gebied.

5.1 De afkomst van MSX DOS

Microsoft heeft al jarenlang een befaamdheid opgebouwd met het ontwikkelen van BASIC-dialecten voor diverse computersystemen. Het Microsoft Basic is voor zeer vele mensen een begrip. Voor zeer vele hobbycomputer-fabrikanten ontwikkelde Microsoft de bijbehorende BASIC-dialecten en baseerde zich hierbij steeds op de centrale taal, het Microsoft basic. Ook het MSX basic is hiervan afgeleid.

Toen enige tijd geleden de 16-bits computers in zwang kwamen, ontwikkelde Microsoft voor dit type computer een eigen operating system; het MS DOS (MicroSoft Disk Operating System). Met dit operating system stak Microsoft haar tegenpool, Digital Research, voor het eerst op operating system gebied de loef af. Tot dan toe gold Digital Research op 8-bits computers als de toonaangevende operating system fabrikant. Het operating system van Digital Research, CP/M (Control Program for Microprocessors) is echter, vooral op het gebied van de 8-bits computers, nog steeds een zeer wijdverbreid operating system. Onder CP/M is nog steeds een gigantische hoeveelheid software op verschillende gebieden te verkrijgen.

Het MSX-DOS is een afgeleide van het MS DOS, speciaal bedoeld voor MSX computers. MSX-DOS is dus op professionele leest geschoeid.

Echter, Microsoft ging bij het samenstellen van MSX-DOS nog een stap verder. Door de struktuur van het KERNEL voor een belangrijk gedeelte gelijk te houden aan de struktuur van het kernel van CP/M schiep Microsoft de mogelijkheid om de gigantische hoeveelheid onder CP/M verkrijgbare, professionele software ook onder MSX-DOS te gebruiken. Vaak moeten hiervoor door de software-fabrikanten kleine wijzigingen worden aangebracht; een enkele keer werkt een CP/M programma onmiddellijk onder MSX-DOS.

Met het MSX-DOS operating system krijgt de MSX computer dus plotseling een toegang tot het immense gebied van CP/M software...

6 HET NUT VAN EEN OPERATING SYSTEM

In de vorige hoofdstukken behandelden we het begrip operating system. Nu we precies weten wat een operating system is, is het misschien zinvol om in te gaan op het nut van de aanwezigheid van het MSX DOS op de MSX computer.

6.1 Compatibiliteit

Compatibiliteit is een woord dat zoveel wil zeggen als: onderlinge toepasbaarheid. Indien twee computers met betrekking tot een programma compatible zijn, betekent dat dat het betreffende programma op beide computers zonder wezenlijke wijzigingen kan worden uitgevoerd. De mate van compatibiliteit wordt bepaald door het aantal veranderingen dat moet worden aangebracht op een programma om het ook op een ander systeem te kunnen laten draaien. Een hoge compatibiliteit houdt in, dat slechts een enkele wijziging mogelijk moet worden aangebracht.

Door gebruik te maken van een kernel met een vaste, afgesproken struktuur, schept men de mogelijkheid om programma's samen te kunnen stellen die op verschillende computersystemen kunnen werken. Voor de daadwerkelijke besturing van de computeronderdelen (randapparaten) worden immers de kernel-routines gebruikt. Deze kernel-routines kunnen per merk computersysteem verschillen; het programma dat deze kernel gebruikt, kan voor elke systeem gelijk blijven.

Zo is bijvoorbeeld software die geschreven is voor het CP/M operating system, op honderden verschillende computersystemen te gebruiken. Al deze computersystemen maken namelijk gebruik van het CP/M operating system en hebben elk een eigen kernel dat machine-afhankelijk maar altijd overeenkomstig van struktuur is.

Het voordeel hiervan is, dat software, geschreven voor de ene machine, ook op een andere machine kan worden gebruikt. Software-fabrikanten kunnen op deze wijze verzekerd zijn van een grote schare van afnemers en als zodanig zonder veel risico software gaan ontwikkelen. Voor de gebruiker heeft dit weer het voordeel dat het aanbod van software bijzonder goed van kwaliteit en bijzonder ruim van afmeting is.

We kunnen over het algemeen dus stellen dat gebruik van een operating system, vooral van een standaard veel gebruikt operating system, direkt

leidt tot een groot en kwalitatief uitstekend software-aanbod. Doordat het aanbod zo hoog is en de fabrikanten kunnen rekenen op een behoorlijke afzet, blijven de prijzen meestal erg vriendelijk.

6.2 MSX-DOS en CP/M

We zagen reeds dat het CP/M operating system nog steeds een zeer wijdverbreid operating system is. In de afgelopen jaren is er een immens aanbod aan software ontstaan voor CP/M computers.

Omdat het MSX-DOS op kernel-niveau een hoge mate van compatibiliteit heeft met CP/M, is de meeste van deze software na hoogstens een kleine aanpassing bruikbaar onder MSX DOS.

Een bijzonder sterk en fraai voordeel van MSX-DOS is dus, dat dit operating system een deur opent naar een reeds bestaande, immens grote hoeveelheid professionele software. Zo kan de MSX-DOS gebruiker kiezen tussen diverse, zeer professionele tekstverwerkers of databases. Ook bedrijfssoftware (financiële administraties, faktureerprogramma's, voorraadcontroleprogramma's, etcetera) is te kust en te keur te verkrijgen. Ook de semi-professionele gebruiker kan met zijn MSX systeem dus alle kanten uit.

6.3 Machine-onafhankelijkheid

Professionele software blijft duur. Hoe de concurrentieverhoudingen ook liggen; voor een goede database moet altijd nog een bedrag van enkele honderden guldens worden neergeteld. Nog veel duurder is natuurlijk de maatsoftware die men speciaal voor een bepaald doel door een software-fabrikant laat ontwikkelen.

Het is dan ook altijd een pijnlijke zaak om op een gegeven moment te moeten constateren dat men bij aanschaf van een nieuw, groter computersysteem verplicht is om ook totaal nieuwe software aan te schaffen. Niet alleen het aanschafbedrag vormt een struikelblok; ook de bediening van die andere software, het aanpassen van reeds aanwezige gegevens en dergelijke zaken vormen een probleem.

Wanneer men gebruik maakt van operating system dat algemeen in zwang is, vallen veel bezwaren weg. Doordat bij verandering van computer het operating system gelijk blijft, behoeft geen nieuwe software te worden aangeschaft maar behoeft de oude software slechts naar het nieuwe systeem te worden overgebracht.

Over het algemeen kan men dus stellen dat gebruik van een operating system er zorg voor draagt, dat bij uitbreiding of vervanging van apparatuur de software behouden kan blijven.

In de voorgaande hoofdstukken behandelden we uitgebreid het onderwerp operating systems in het algemeen en MSX-DOS in het bijzonder.

In de volgende hoofdstukken gaan we in op de praktijk van het MSX DOS.

7.1 Volgorde van handelingen

Om het MSX-DOS operating system op te starten, hebben we de volgende dingen nodig:

- een MSX computer
- een aangekoppelde schijveneenheid (geen Quick Disk)
- een schijfje met daarop het MSX-DOS

Het MSX-DOS is als volgt op te starten:

- plaats het schijfje met het MSX DOS operating system in de schijveneenheid. Wanneer de betreffende schijveneenheid een vergrendeling heeft, laat deze dan nog even open. Wanneer geen vergrendeling aanwezig is, plaats het schijfje dan niet helemaal maar laat het nog iets uitsteken.

- schakel vervolgens de schijveneenheid in.

- schakel vervolgens de MSX computer (en beeldscherm) in.

- vergrendel nu onmiddellijk de schijveneenheid of plaats de schijf volledig in de schijveneenheid.

Na enige seconden gaat de schijveneenheid werken en vrij snel volgt op beeld de melding:

```
MSX-DOS version 1.00
Copyright 1984 by Microsoft

Command version 1.01

Current date is Sun 1-01-1984
Enter new date:
```

```
Current time is 12:00:00a
Enter new time:
```

Door twee maal alleen de RETURN-toets in te geven, slaan we voorlopig de ingave van datum en tijd over. Zie voor de juiste ingaven de behandeling van het kommando DATE en TIME in hoofdstuk 8.

Sommige systemen vragen alleen naar de datum, niet naar de tijd.

Uiteindelijk verschijnt de melding:

```
A>
```

Deze melding geeft aan welke schijveneenheid op dit moment aktief is. De letter A, meestal de PROMPT genoemd, staat voor de eerste aangekoppelde schijveneenheid, de letter B voor de tweede, enzovoorts.

Het >-teken geeft aan dat we een kommando kunnen ingeven.

Wat we zojuist deden, is het opstarten van het MSX DOS operating system. Dit opstarten wordt ook wel het 'booten' van het systeem genoemd. We onderscheiden de zogenaamde COLD BOOT (koude start) waarbij het systeem wordt opgestart door het aan te zetten en de WARM BOOT waarbij het systeem hernieuwd wordt opgestart zonder het systeem eerst uit te zetten.

Wanneer een gebruikersprogramma ten einde is en het controlerend programma weer binnenhaalt, voert dit programma een zogenaamde WARM BOOT uit. Door middel van het tegelijk indrukken van de toetsen CTRL en C (of CTRL en STOP) forceert men eveneens een WARM BOOT; de prompt wordt hernieuwd afgedrukt.

7.2 De eerste, eenvoudige handelingen

Nu het operating system met succes is opgestart, kunnen we de eerste eenvoudige besturing met behulp van dit operating system uitvoeren.

Eén van de meest eenvoudige besturingen is het veranderen van aktieve schijveneenheid. Na een COLD BOOT is altijd schijveneenheid A aktief. Wil men echter schijveneenheid B aktief maken, dan kan dat door ingave van:

```
B:
```

Het systeem reageert meteen met de prompt:

```
B>
```

als teken dat de tweede schijveneenheid nu aktief is. Zo kunnen in principe zesentwintig schijveneenheden met alle letters van het alfabet worden geaktiveerd. Deze besturingsmogelijkheid is meestal zinloos bij het gebruik van slechts één schijveneenheid.

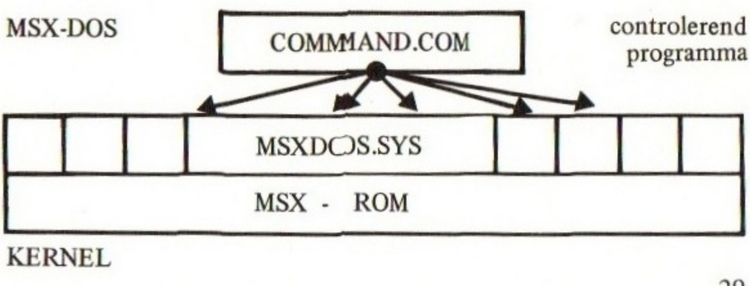
De inhoud van een schijf kan eenvoudig als volgt worden opgevraagd:

```
B>A:                  (de eerste schijveneenheid weer aktief maken)
A>DIR                 (afkorting van DIRECTORY=inhoudsopgave)
MSXDOS    SYS    2432   1-01-1984
COMMAND   COM    6272   1-01-1984
       2 files   173215 bytes free
A>
                                    (de inhoud wordt weergegeven)
```

In het bovenstaande voorbeeld zijn de inhoud en de getalswaarden willekeurig; op uw systeem kunnen hele andere bestanden, andere data en andere grootten worden gegeven. Eén ding staat echter vast; twee bestanden staan in ieder geval in de zojuist opgevraagde directory:

MSXDOS.SYS en COMMAND.COM

In deze twee bestanden is het volledige operating system opgenomen. In hoofdstuk 4 behandelden we de opbouw van een operating system; welnu, het in dat hoofdstuk geschetste schema kunnen we als volgt invullen:



De werkelijke kernel van MSX-DOS staat voor het grootste gedeelte in het ROM (read only memory) van de MSX computer. Van ditzelfde kernel maakt ook het MSX basic bijvoorbeeld gebruik.
Het bestand MSX-DOS. SYS is een aanvulling op dit kernel. Met name maakt dit gedeelte van het kernel het ROM-geheugen van de MSX computer bereikbaar voor MSX-DOS.

Het bestand COMMAND.COM is het zogenaamde controlerende programma. Dit programma staat toe dat het vervangen wordt door een ander programma. Ook maakt het het eenvoudige systeemonderhoud (bestanden veranderen/aanmaken/verwijderen, systeemdatum veranderen etc.) mogelijk.

We zien dat deze twee bestanden meteen voorbeelden zijn van een SYSTEM FILE en een COMMAND FILE (hoofdstuk 3).

Dat COMMAND.COM een COMMAND FILE is, kunnen we aantonen door de volgende proefneming. Tik in:

```
A>COMMAND
```

De schijveneenheid gaat even werken waarna de prompt weer verschijnt. De opdracht COMMAND resulteerde hierin, dat het controlerend programma (COMMAND.COM) werd vervangen door een andere COMMAND FILE. Deze andere command file bleek hetzelfde controlerende programma te zijn;COMMAND.COM werd slechts herladen en opnieuw in werking gesteld. In feite voerden we zojuist een WARM BOOT uit.

8 BESTANDSAANDUIDINGEN

Het MSX-DOS is een DISK operating system. De onderhoudsakties die via het MSX-DOS kunnen worden uitgevoerd, bestaan dan ook hoofdzakelijk uit akties met betrekking tot de schijveneenheid. Zo kunnen bijvoorbeeld de volgende akties standaard onder MSX-DOS worden ontplooid:

– het kopiëren van bestanden
– het veranderen van bestandsnamen
– het verwijderen van bestanden
– het opvragen van bestandsspecificaties.

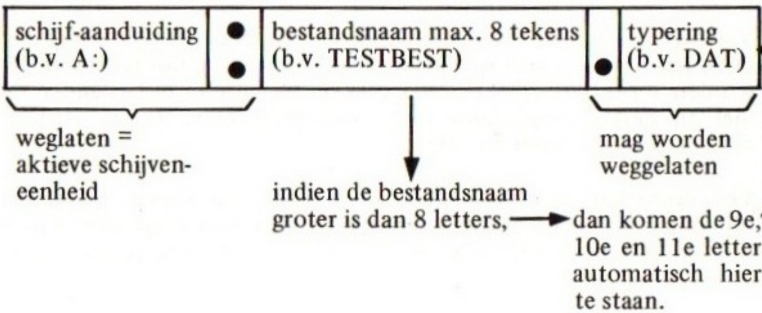
Om bij het onderhoud van het MSX-DOS systeem aan te duiden om welk bestand of om welke groep van bestanden het gaat, kennen we het begrip:

⟨BESTANDSAANDUIDING⟩

Een bestandsaanduiding is een naamgeving waarmee we geen, één of een groep van bestanden kunnen aanduiden. Dit in tegenstelling tot een bestands*naam*, waarmee we precies één bestand kunnen aanduiden.

Om verder op deze term in te gaan, dienen we ons eerst te realiseren welke componenten een bestands*naam* op schijf in zich heeft of in zich kan hebben:

bestandsnaam: op schijf (b.v. A:TESTBEST.DAT)



Allereerst onderscheiden we in de bestandsnaam de schijfaanduiding (A:/B:/enz.).
Vervolgens onderscheiden we de naam van het bestand.
Achter de naam mag vervolgens een punt worden opgenomen, gevolgd door een drie-letterige typeaanduiding. Deze type-aanduiding is geheel vrij en mag naar believen worden toegepast. Het is echter raadzaam om een vaste regel in verband met deze typering te hanteren. Bijvoorbeeld:

– een basic programma op schijf (geen ,A)	XXXXXXXX.BAS
– een basic programma op schijf (met ,A)	XXXXXXXX.ASC
– een bestand met daarin alleen tekst	XXXXXXXX.TXT
– een sequentiëel bestand	XXXXXXXX.SEQ
– een random bestand	XXXXXXXX.RAN

Door alle bestanden van een bepaalde soort ook een bepaalde typeaanduiding te geven, zijn deze later (wanneer er tientallen bestanden op schijf staan) weer snel bij elkaar te zoeken.

Schijf-aanduiding en typering mogen worden weggelaten. Indien de schijf-aanduiding wordt weggelaten, wordt de laatst geaktiveerde schijveneenheid aangenomen.

In het MSX-DOS is het bij sommige kommando's noodzakelijk om een groep van bestanden tegelijk te kunnen aanduiden. Dit doen we met een zogenaamde bestandsaanduiding.

Een bestandsaanduiding kan allereerst een schijf-aanduiding bevatten (bijvoorbeeld A: of B:) maar dit is niet noodzakelijk.

Vervolgens kan de bestandsaanduiding een bestandsnaam bevatten. Deze naam mag geheel worden ingevuld, maar sommige tekens mogen worden vervangen door een vraagteken. De groep van bestanden die met die bestandsaanduiding wordt bedoeld, mogen in hun naam op deze positie *elke* letter bevatten.

Vervolgens kan een type-aanduiding worden opgenomen. Voor deze typering geldt eveneens dat sommige tekens door een vraagteken mogen worden vervangen. Ook mag een typering geheel worden weggelaten.

Enkele voorbeelden:

bestandsaanduiding	groep van bestanden die wordt aangeduid
"BES??"	alle bestanden op de laatst geaktiveerde schijveneenheid die een naam van 5 letters hebben waarvan de eerste drie letters gelijk zijn aan BES
"A:?FILE"	alle bestanden op schijveneenheid A: (de eerste schijveneenheid) met een naam van 5 letters waarvan de laatste vier letters gelijk zijn aan FILE
"A:????.BAS"	alle bestanden op schijveneenheid A: met een naam van vier letters en een typering BAS
A:PROGRAMM.???	alle bestanden op schijveneenheid A: met de naam PROGRAMM en een typering van drie letters

Zowel de bestandsnaam als de typering mogen als *laatste* karakter een sterretje bevatten. Met dit sterretje geven we aan, dat de rest van de bestandsnaam of de typering er niet toe doet. Bijvoorbeeld:

bestandsaanduiding	groep van bestanden die wordt aangeduid
"BES*.DAT"	alle bestanden met een naam die begint met BES. De lengte van de naam doet er verder niet toe. Wel dient het bestand een typering DAT te hebben. De bestanden dienen aanwezig te zijn op de laatst geaktiveerde schijveneenheid
"*.P*"	alle bestanden met een typering, beginnende met een P op de laatst geaktiveerde schijf
"A:*.*"	alle bestanden op schijveneenheid A:

Uiteraard mag een bestandsaanduiding ook gewoon een bestandsnaam bevatten. De 'groep' van bestanden die wordt bedoeld, bevat dan altijd maar één bestand.

System files, batch files en command files kennen (zie hoofdstuk 3) een eigen, aparte typering. Het is raadzaam om deze typeringen alleen voor de bijbehorende bestandssoorten te gebruiken. De typeringen zijn als volgt:

– een command file	XXXXXXXX.COM	(bijvoorbeeld COMMAND.COM)
– een system file	XXXXXXXX.SYS	(bijvoorbeeld MSXDOS.SYS)
– een batch file	XXXXXXXX.BAT	

NB: in bestandsnamen mogen de volgende tekens worden gebruikt:

```
A-Z     0-9     $     &     #     %     ,

( )     -       @     /     ^     { }   !
```

Indien kleine letters worden gebruikt, zet MSX-DOS deze zelf om in hoofdletters.

9. DE KOMMANDO'S VAN MSX-DOS

9.1 De intrinsieke kommando's van MSX-DOS

Het operating system heeft onder meer tot taak, het onderhoud van het betreffende computersysteem mogelijk te maken. Hiertoe dienen enige bevelen aan wezig te zijn waarmee dit onderhoud kan geschieden.

De direkt bij het operating system betrokken kommando's die niet zondermeer van dit operating system zijn los te maken, noemt men de intrinsieke kommando's van dat operating system.

In deze paragraaf worden de intrinsieke kommando's van MSX-DOS stuk voor stuk behandeld.

Per kommando is een schrijfwijze, een toelichting op de betekenis en een voorbeeld opgenomen.

De schrijfwijze dient als volgt te worden gelezen:

Met normale hoofdletters worden de gegevens aangegeven die direkt dienen te worden ingetoetst.

Tussen schuine haken (⟨en⟩) worden de gegevens aangegeven die niet direkt dienen te worden ingetoetst. De tekst tussen schuine haken omschrijft slechts het gegeven dat dient te worden ingetoetst.

Eén zo'n gegeven dat we veelvuldig tegen zullen komen en dat tussen schuine haken wordt genoteerd, is de zogenaamde

<BESTANDSAANDUIDING>

die in het vorige hoofdstuk reeds werd behandeld.

De tussen vierkante haken genoteerde gegevens ([en]) mogen eventueel zonder verplichting worden toegepast.

schrijfwijze

```
BASIC [<PROGRAMMANAAM>]
```

betekenis

Dit kommando is het eenvoudigste MSX-DOS kommando. Door eenvoudigweg na de schijfaanduiding het woord:

```
A>BASIC
```

in te toetsen, verschijnt de MSX-BASIC melding en is het MSX-BASIC aktief geworden.

Door onder MSX-BASIC de volgende CALL in te toetsen:

```
CALL SYSTEM
```

kan vanuit MSX-BASIC het MSX-DOS weer worden geactiveerd.

Het kommando BASIC zorgt ervoor, dat het 32 kilobytes grote ROM-geheugen (ROM=read only memory, onveranderbaar, alleen leesbaar geheugen) waarin het MSX-BASIC ligt opgeslagen, wordt geactiveerd. Hiertoe dient de eerste 32 kilobytes aan RAM-geheugen (RAM=random access memory, vrij toegankelijk geheugen), dat onder MSX-DOS actief is, te worden afgekoppeld. Bij CALL SYSTEM gebeurt natuurlijk precies het omgedraaide.

Wanneer achter het kommando BASIC een programmanaam wordt opgegeven, dan wordt het betreffende basic-programma onmiddellijk na het activeren van het MSX-BASIC uitgevoerd. Het programma EDITOR.BAS dat in dit boek is opgenomen, kan vanuit MSX-DOS onmiddellijk worden opgestart met het kommando:

```
A>BASIC EDITOR.BAS
```

schrijfwijze

```
FORMAT
```

Met het FORMAT-kommando kunnen schijven worden geformatteerd. De betekenis en noodzaak van dit formatteren kwamen we reeds tegen in hoofdstuk 2. Daar zagen we ondermeer dat een nieuwe schijf altijd eerst een keer moet worden geformatteerd voordat deze kan worden gebruikt.

Pas op: een schijf verliest bij het formatteren alle daarop aangebrachte gegevens. Een gouden regel is dan ook om, hoe zeker u ook van uw zaak bent, voordat u daadwerkelijk gaat formatteren, twee maal te controleren of u de juiste schijf in de betreffende eenheid heeft geplaatst. Indien u meerdere schijfeenheden heeft, ontgrendel dan de overige eenheden zodat een eventueel daarin geplaatste schijf niet per ongeluk kan worden geformatteerd.

Nadat het kommando FORMAT is ingegeven, vraagt de computer om een specificatie van de schijveneenheid waarin de te formatteren schijf is geplaatst. Geef voor de eerste schijveneenheid de letter A in, voor de tweede schijveneenheid de letter B, enzovoorts. Indien u maar één schijveneenheid heeft, geeft u het beste altijd een A in.

Het formatteren van een schijf kan enkele minuten duren. Uiteindelijk geeft de computer de melding FORMAT COMPLETE wanneer het formatteren naar behoren is voltooid. Wanneer een fout bij het formatteren optreedt, dient het formatteren nog een keer te worden uitgevoerd. Een schijf die na drie maal formatteren nog niet met de juiste melding FORMAT COMPLETE komt, is niet betrouwbaar en dient te worden vervangen. Indien een foutmelding veelvuldig terugkomt, is het misschien tijd om de lees/schrijfkoppen van de betreffende schijveneenheid te (laten) schoonmaken. Hiertoe zijn speciale schoonmaaksets in de handel.

Uiteraard kan het geven van foutmeldingen bij het formatteren ook een technische oorzaak hebben. Het kan zijn dat de betreffende eenheid stuk is of de verkeerde schijven worden gebruikt.

Wanneer u de voorbeelden uit de rest van deze paragraaf wilt bestuderen, is het raadzaam om enige geformatteerde schijven te hebben klaarliggen. Geeft u hiertoe in:

```
A>FORMAT
```

De computer vraagt u onmiddellijk

```
Drive Name? (A,B)
```

Plaatst u op dat moment een nieuw, te formatteren schijf in de eenheid en toets een A (of een B) in.

De computer vraagt u vervolgens:

```
Strike a key when ready
```

Geef ter bevestiging een willekeurige toets in en wacht op de melding:

```
Format complete
```

en herhaal deze behandeling voor alle andere te formatteren schijven.

Wanneer u per ongeluk het FORMAT-kommando heeft ingetoetst en u wilt geen schijven initialiseren, druk dan de CTRL en de C-toets (of de CTRL en de STOP-toets) in. Het formatteren wordt afgebroken.

schrijfwijze

```
DIR [<PROJEKTIEWIJZE>] [<BESTANDSAAN-
DUIDING>] [<PROJEKTIEWIJZE>]
```

betekenis

Met het DIR-kommando kunnen we een inhoudsopgave van een schijf op beeldscherm verkrijgen. Door bijvoorbeeld alleen het volgende kommando in te geven:

```
A>DIR
```

verkrijgen we een inhoudsopgave van de schijf die het laatste werd geactiveerd; in dit geval schijf A.

Door ingave van:

```
A>DIR *.TXT
```

verschijnt op beeldscherm een opgave van alle bestanden op schijf A met typering TXT.

Eventueel kunnen we bij DIR een projektiemethode toepassen. Er zijn twee verschillende projektiemethoden mogelijk, namelijk:

/P — de letter P staat voor page of bladzijde. Na elke 'bladzijde', steeds als het scherm opnieuw gevuld is zal de computer niet gewoon doorgaan met het projekteren van de inhoud maar zal eerst om een extra toetsaanslag worden gevraagd. Deze projektiemethode is alleen van nut bij schijven die betrekkelijk veel bestanden bevatten.

/W — de letter W staat voor wide. Wanneer we deze projektiemethode opgeven, krijgen we de inhoud op beeldscherm voorgeschoteld in verkorte vorm over twee kolommen. De gegevens omtrent de grootte van de bestanden en de datum waarop deze bestanden werden aangelegd, worden in dat geval niet getoond.

De verschillende projektiemethoden mogen indien van toepassing zowel voor als achter de bestandsaanduiding worden opgenomen en mogen in elke volgorde met elkaar worden gekombineerd.

Enkele voorbeelden:

`DIR /P`	geeft de inhoudsopgave van de laatst geactiveerde schijf. Wanneer het beeldscherm vol is, vraagt de computer u om een willekeurige toetsingave waarna verder wordt geprojekteerd.
`DIR /W B:*.BAS`	De inhoud van schijf B wordt in gecomprimeerde vorm getoond. Alleen de bestanden met typering BAS worden getoond.
`DIR A:A??.*/P/W`	Alle bestanden op de schijf in de eerste eenheid met een bestandsnaam van drie letters waarvan de eerste letter gelijk is aan de letter A, worden getoond. Eventueel wordt na projektie van een vol beeldscherm gewacht totdat een toets wordt ingedrukt. De inhoudsopgave wordt weer in verkorte vorm weergegeven.

Het volgende voorbeeld geeft de inhoudsopgave van zo maar een schijf:

```
>DIR
MSXDOS   SYS    2560  1-01-84
TEST     BAS     154  1-01-84
COMMAND  COM    6400  1-01-84
EDITOR   BAS    5494  1-01-84
EDITOR   BAT      15  6-05-85
AUTOEXEC BAT      84  1-01-84
BYTES    BAT      14  7-05-85
        7 files   135680 bytes free
```

In de eerste kolom zien we de bestandsnamen genoemd. In de tweede kolom worden de typeringen opgesomd.
In de derde kolom zien we de grootte van elk bestand in aantal bytes uitgedrukt terwijl in de vierde kolom de data zijn opgenomen waarop de betreffende bestanden zijn toegewezen.
Uiteindelijk wordt aan het einde van de inhoudsopgave het totaal aantal afgedrukte bestandsnamen opgegeven alsmede het aantal nog vrije bytes op de schijf.

schrijfwijze

```
MODE <BEELDSCHERM>
```

betekenis

Met het MODE-kommando kan de breedte van het beeldscherm in aantal tekens worden opgegeven. Met het kommando:

```
MODE 40
```

kan de beeldschermbreedte bijvoorbeeld op 40 karakters worden gezet.

Wanneer een beeldschermbreedte van 32 karakters of minder wordt gespecificeerd, dan gaat MSX-DOS in beeldscherm mode 1 over (zie MSX-BASIC handboek). Boven de 32 karakters gaat MSX-DOS over op beeldscherm mode 0.

De opgegeven breedte dient altijd een geheel getal te zijn, groter dan 0 en kleiner dan 41.

schrijfwijze

```
PAUSE [<KOMMENTAAR>]
```

Het PAUSE-kommando geeft de melding "Strike any key when ready..." en wacht totdat een toets wordt ingedrukt. Het achter het kommando PAUSE opgenomen kommentaar wordt verder niet in behandeling genomen en is alleen bedoeld als aanwijzing voor de lezer van het beeldscherm.

In het gewone gebruik heeft het PAUSE-kommando geen nut. In hoofdstuk 10 behandelen we het begrip BATCH-FILES. In dat hoofdstuk krijgt dit bevel plotseling groot nut.

schrijfwijze

```
REM [<KOMMENTAAR>]
```

betekenis

Het kommando REM is een afkorting van REMARK (opmerking). Net als in BASIC staat het REM-kommando ons toe, een verduidelijking of kommentaar op te nemen. Een verschil is hierin gelegen dat MSX-DOS de REM-kommando's ook afdrukt.

Bij het gewone gebruik heeft het REM-kommando weinig waarde. Wat voor zin heeft het immers om een opmerking op het beeldscherm in te toetsen zonder dat er verder ooit nog iets mee gedaan wordt?

In hoofdstuk 10 behandelen we het begrip BATCH-FILES. In dat hoofdstuk krijgt het REM-kommando pas zijn waarde wanneer we programma-achtige strukturen in MSX-DOS gaan ontwerpen.

schrijfwijze

```
TYPE <BESTANDSAANDUIDING>
```

betekenis

Het kommando TYPE probeert voor u de inhoud van het door u aangegeven bestand op beeldscherm weer te geven. Wanneer het betreffende bestand alleen leesbare karakters bevat (een zogenaamde ASCII-file of textfile), dan lukt dit zonder problemen. Wanneer u echter een ander bestand (een BASIC-programma of iets dergelijks) met TYPE probeert te onderzoeken, wordt het op het beeldscherm meestal een enorme ravage. Doordat het betreffende bestand eigenlijk niet geschikt is om te worden afgedrukt, worden de gegevens vaak kris kras door elkaar afgedrukt. Veel van de karakters die TYPE probeert af te drukken, zijn geen zonder meer afdrukbare karakters en veroorzaken bij het projekteren vreemde effecten. Probeer dit maar eens met:

```
A>TYPE COMMAND.COM
```

Een tekstbestand, bijvoorbeeld een met de A-optie op schijf geplaatst programma, kan met TYPE zonder problemen op beeldscherm worden afgedrukt.

Indien geen vaste bestandsnaam maar een bestandsaanduiding aan kommando TYPE is opgegeven, dan wordt het eerste bestand uit de inhoudsopgave dat aan de opgegeven bestandsaanduiding voldoet, afgedrukt.

Voorbeeld: (bij voorkeur uit te voeren op een nieuwe, geformatteerde schijf)

Met het COPY-kommando, dat verderop wordt behandeld, kunnen we ondermeer eenvoudig een tekst in een bestand vastleggen:

```
A>COPY CON TEST.TXT
DEZE TEKST WORDT STRAKS WEER
MET HET TYPE-KOMMANDO OP HET
BEELDSCHERM GEPROJEKTEERD.
^Z
        1 file copied
```

De ingave beëindigen we door intoetsing van tegelijk de CTRL en de Z-toets en daarna de RETURN-toets.

Met het TYPE-kommando kunnen we de vastgelegde tekst dan weer afdrukken:

```
A>TYPE TEST.TXT
DEZE TEKST WORDT STRAKS WEER
MET HET TYPE-KOMMANDO OP HET
BEELDSCHERM GEPROJEKTEERD.
A>
```

schrijfwijze

```
DEL <BESTANDSAANDUIDING>
of
ERASE <BESTANDSAANDUIDING>
```

betekenis

Met het kommando ERASE of DEL (beide doen precies hetzelfde) kunnen bestanden van schijf worden verwijderd. Eén bestand maar ook groepen van bestanden kunen tegelijk worden verwijderd, afhankelijk van de opgegeven bestandsaanduiding.

Voorbeelden:

`DEL TEST.TXT`	verwijdert bestand TEST.TXT op de laatst geactiveerde schijf
`ERASE A:*.*`	verwijdert ALLE bestanden op de schijf in de eerste schijveneenheid
`DEL A:*.TXT`	verwijdert alle bestanden van schijf A met typering TXT
`DEL B:M??.*`	verwijdert van de schijf in schijveneenheid nummer 2 alle bestanden die een naam hebben van drie letters, beginnende met een M.

schrijfwijze

```
REN <BESTANDSAANDUIDING OUD> <BE-
    STANDSAANDUIDING NIEUW>
of
RENAME <BESTANDSAANDUIDING OUD>
       <BESTANDSAANDUIDING NIEUW>
```

betekenis

Met het kommando REN of RENAME (hebben dezelfde funktie) kan een bestand of kunnen groepen van bestanden een andere naam worden gegeven. Hiertoe dient achter het REN(AME)-kommando de oude bestandsaanduiding te worden opgenomen, gevolgd door de nieuwe bestandsaanduiding.

Enkele voorbeelden:

`REN TEST.BAS TEST.BAK`	bestand TEST.BAS krijgt de nieuwe naam TEST.BAK
`RENAME *.BAS *.TXT`	alle bestanden met typering BAS krijgen nu de typering TXT
`REN *.* Q*.*`	alle bestandsnamen worden zo veranderd dat de eerste letter een Q wordt
`RENAME ???.TXT ???`	alle bestanden met een naam van drie letters en een typering TXT behouden hun eigen naam echter zonder typering.
`REN FILE1 ????2`	bestand FILE1 krijgt de nieuwe naam FILE2

```
RENAME A:FILE B:FILE
```

deze RENAME-opdracht resulteert onmiddellijk in een fout; een bestand kan natuurlijk niet naar een andere schijf hernoemd worden.

Wanneer REN of RENAME uitvoering het gevolg zou hebben dat er twee of meer bestanden met een zelfde naam op een zelfde schijf ontstaan, wordt de melding 'Rename error' gegeven en wordt de opdracht verder onderbroken.

Het volgende voorbeeld laat u een inhoudsopgave van een schijf, een RENAME-opdracht en weer een inhoudsopgave van dezelfde schijf na deze opdracht zien:

```
A>DIR
TEST01    TXT     3200  6-16-85
TEST02    TXT     1235  6-16-85
WS        COM     2230  5-12-84
SC        COM    12004  5-12-84
EDITOR    BAS     7192  6-17-85
TEST05    TXT      122  6-17-85
      6 files   172305 bytes free
A>RENAME TEST??.TXT PROG??.TXT
A>DIR
PROG01    TXT     2300  6-16-85
PROG02    TXT     1235  6-16-85
WS        COM     2230  5-12-84
SC        COM    12004  5-12-84
EDITOR    BAS     7192  6-17-85
PROG05    TXT      122  6-17-85
      6 files   172305 bytes free
A>
```

schrijfwijze

```
DATE [<DATUMOPGAVE>]
```

betekenis

Met het DATE-kommando kan de systeemdatum van de computer worden veranderd. Standaard staat deze datum op 1 januari 1984. Bij het opstarten van een MSXDOS-schijfje wordt, tenzij bestand AUTO-EXEC.BAT op de schijf staat (zie hoofdstuk 10), het DATE-kommando automatisch uitgevoerd.

Achter het DATE-kommando kan onmiddellijk een datum worden opgegeven. Deze datum dient als volgt te zijn opgebouwd:

MM.DD.JJ of MM/DD/JJ of MM-DD-JJ

of

MM.DD.JJJJ of MM/DD/JJJJ of MM-DD-JJJJ

MM stelt hierbij het nummer van de maand voor en mag variëren van 1 tot 12
DD stelt het dagnummer in de maand voor en mag variëren van 1 tot 31
JJ stelt het jaarnummer voor en mag variëren van 0 tot 99
JJJJ stelt het volledige jaar voor en mag variëren van 1980 tot 2099

Wanneer alleen de laatste twee posities van het jaar worden gespecificeerd, dan neemt het MSX-DOS-systeem het volgende aan;

JJ=80-99	aangenomen wordt het jaar 1980-1999
JJ=0-79	aangenomen wordt het jaar 2000-2079

Het DATE-kommando kontroleert op de juiste datum en houdt daarbij ook rekening met schrikkeljaren. De datum 2/29/85 wordt bijvoorbeeld niet geaccepteerd terwijl de datum 2/29/84 wel wordt geaccepteerd.

Wanneer alleen het DATE-kommando, zonder achtergevoegde datum, wordt uitgevoerd, dan geeft het systeem de interne datum op beeldscherm weer. Met alleen de RETURN-toets kan de daaropvolgende ingave worden overgeslagen.

In het volgende voorbeeld wordt eerst de datum op 29 februari 1984 gesteld. Vervolgens wordt deze datum weer opgevraagd. Merk op dat MSX-DOS keurig de juiste dag bij de datum vermeldt.

```
A>DATE
Current date is Sun  1-01-1984
Enter new date: 02/29/84

A>DATE
Current date is Wed  2-29-1984
Enter new date:
```

schrijfwijze

```
TIME [<TIJDSOPGAVE>]
```

Met het TIME-kommando kan de tijd van de MSX-computer worden gelijkgezet. Standaard staat de tijd bij het opstarten altijd op twaalf uur middernacht precies. Bij het opstarten van een MSX-DOS-schijfje wordt het TIME-kommando op de meeste systemen standaard uitgevoerd. Er zijn echter systemen waarbij het TIME-kommando geen enkele zin heeft.

Achter het TIME-kommando kan onmiddellijk een tijd worden ingevoerd. Deze tijd dient als volgt te zijn opgebouwd:

UU of UU:MM of UU:MM:SS

of

UUP of UU:MMP of UU:MM:SSP

Zonder achtervoegsel of met het achtervoegsel P mag dit gegeven variëren van 0 tot 23. Met achtervoegsel A mag dit gegeven variëren van 0-12. Het achtervoegsel A geeft aan dat het om een tijd gaat tussen 12 uur 's nachts en 12 uur 's middags. Het achtervoegsel P geeft aan dat het om een tijd gaat tussen 12 uur 's middags en 12 uur 's nachts. De tijd kan dus volgens een 12 uurs klok en een 24 uurs klok worden ingegeven.

MM staat voor het minuut-gegeven van de tijd en mag variëren van 0 tot 59.

SS staat voor het seconden-gegeven van de tijd en mag eveneens variëren van 0 tot 59.

Enkele voorbeelden:

`TIME 12`	de tijd wordt op 12 uur 's middags gezet
`TIME 23:59:59`	de tijd wordt op één seconde voor middernacht gezet
`TIME 11:30P`	de tijd wordt op half twaalf 's avonds gezet

Wanneer alleen het kommando TIME wordt ingetoetst, vermeldt het MSX-DOS eerst de oude tijd en vraagt dan om een eventuele nieuwe tijd. Deze nieuwe ingave kan met alleen de RETURN-toets worden overgeslagen waarmee de oude tijd actief blijft.

schrijfwijze

```
VERIFY ON
of
VERIFY OFF
```

betekenis

Met het VERIFY-kommando kan het MSX DOS worden bevolen, éénmaal op de schijf geschreven gegevens nog éénmaal ter controle onmiddellijk weer in te lezen en te vergelijken. Het resultaat hiervan is natuurlijk dat het schrijven van gegevens op schijf wat langzamer verloopt.

Met VERIFY ON zet men deze controlemogelijkheid aan en
met VERIFY OFF zet men deze controle weer uit.

Standaard staat deze controlemogelijkheid uit.

Sommige toepassingen van het MSX-DOS systeem kennen deze controlemogelijkheid niet. Weliswaar kunt u VERIFY ON en VERIFY OFF ingeven, maar deze ingaven hebben geen enkel resultaat.

schrijfwijze

```
COPY [<KOPIEERAANDUIDING>]
```

betekenis

Het kopiëerkommando is het meest uitgebreide kommando dat onder MSX-DOS bestaat. Hieronder wordt het kopiëerkommando in verschillende fasen behandeld, in opkomende graad van moeilijkheid.

1 het kopiëren van een schijf

Een bestand bevat bepaalde gegevens. Op een moment kan het raadzaam zijn om deze gegevens te kopiëren. Wanneer er met de originele gegevens eventueel iets fout mocht gaan, kunt u in dat geval namelijk altijd nog van de eerder gekopiëerde gegevens uitgaan.

Eén van de belangrijkste dingen die u moet doen nog voordat u serieus bepaalde aktiviteiten op de MSX-DOS schijf gaat uitvoeren, is deze MSX-DOS schijf kopiëren.

Voordat u een gehele schijf gaat kopiëren, moet u eerst zorgen voor een lege, geformatteerde schijf waarop u de gegevens kan kopiëren. Formatteer deze schijf eerst met het FORMAT-kommando.

Daarna gaat, wanneer u de beschikking heeft over slechts één schijveneenheid, het kopiëren van een gehele schijf als volgt:

– plaats de te kopiëren schijf in de schijveneenheid

– geef in: `A>COPY A:*.* B:`

Met dit bevel geven we aan dat alle bestanden van de eerste schijveneenheid (A:*.*) naar de tweede schijveneenheid (B:) moeten worden gekopiëerd. We geven het COPY-kommando dus eerst aan, waarvandaan hij de gegevens moet kopiëren en dan pas waar naartoe deze gegevens dienen te worden gekopiëerd. Omdat één schijveneenheid de rol van twee schijveneenheden kan vervullen onder MSX-DOS, kan op deze wijze een gehele schijf worden gekopiëerd.

De computer laat zien, welke bestanden worden geladen en dadelijk op de nieuwe schijf zullen worden gezet. Wanneer de computer u vraagt:

```
Insert diskette for drive B:
and strike any key when ready
```

Plaats de nieuwe schijf dan in de schijveneenheid en toets een willekeurige toets in.

Wanneer de computer u vraagt:

```
Insert diskette for drive A:
and strike any key when ready
```

plaats dan weer de oude schijf in de schijveneenheid en geef een willekeurige toets in.

Het kan zijn, bij vrij volle schijven, dat de computer u meerdere malen om een schijvenwisseling vraagt. Doet u dit net zo lang totdat het kopiëren ten einde is.

Op deze wijze kunt u meerdere kopieën van uw schijven maken. Het is raadzaam om van belangrijke schijven minstens twee kopieën te bewaren. Door het goed bijhouden van kopieën is het gevaar, belangrijke gegevens (programma's) te verliezen, minimaal.

Bezitters van twee schijveneenheden kunnen op dezelfde manier kopiëren. Zij moeten er echter eerst voor zorgen dat in de eerste schijveneenheid de originele schijf is geplaatst en in de tweede schijveneenheid de geformatteerde schijf is geplaatst. Tijdens het kopiëren vraagt de computer in dit geval niet om schijfwisselingen.

We zien dat MSX-DOS, wanneer er maar één schijveneenheid beschikbaar is, de mogelijkheid heeft om deze schijveneenheid afwisselend de rol van eerste en van tweede schijveneenheid te laten spelen. Op deze wijze kunnen we bij vele kommando's net doen alsof er twee schijveneenheden aan de computer zijn aangesloten.

2 het kopiëren van een enkel bestand

We hoeven binnen MSX-DOS niet verplicht gehele schijven te kopiëren. Wanneer we dat willen, kunnen we ook enkele bestanden kopiëren.

Dit kunnen we doen door alleen de betreffende bestandsnamen aan het COPY-bevel door te geven.

Wanneer we bijvoorbeeld alleen het bestand TEST.TXT naar een andere schijf willen kopiëren, kunnen we dat als volgt doen:

```
A>COPY TEST.TXT B:
```

De computer vraagt op de gebruikelijke wijze op een gegeven moment naar de schijf waarop dit bestand moet worden bijgeplaatst en geeft uiteindelijk de melding 1 file copied.

Wanneer we het bestand TEST.TXT naar een andere schijf willen kopieren en het meteen een andere naam willen geven, kan men dat als volgt doen:

```
A>COPY TEST.TXT B:TEKST.TXT
```

Het bestand TEST.TXT krijgt op de andere schijf nu de naam TEKST. TXT mee; de inhoud van het bestand blijft natuurlijk hetzelfde.

Het kan ook zijn dat we een kopie van een bestand onder een andere naam op *dezelfde* schijf willen hebben. Omdat een bestand op een schijf een unieke naam moet hebben, is het natuurlijk niet mogelijk om twee bestanden onder dezelfde naam op dezelfde schijf te hebben.

Wanneer we bijvoorbeeld het bestand COMMAND.COM onder de naam COMMAND1.COM op dezelfde schijf willen kopiëren, dan kan dat als volgt:

```
A>COPY COMMAND.COM COMMAND1.COM
```

Na een korte tijd geeft de computer een melding dat het bestand is gekopiëerd. Het bestand COMMAND.COM staat nu onder twee verschillende namen op dezelfde schijf. Met het DIR-kommando kan men dat natuurlijk voor de zekerheid even controleren.

3 het kopiëren van groepen van bestanden

We zagen in de bovenstaande behandeling reeds dat het COPY-kommando steeds twee aanduidingen verlangt. De eerste aanduiding geeft aan, welk bestand of welke bestanden dienen te worden gekopiëerd terwijl de tweede aanduiding aangeeft, wat de bestemming van de te kopiëren bestanden is.

Zo hebben de volgende kopiëeropdrachten de daarachter vermelde resultaten:

```
A>COPY A:*.* B:
```

kopiëer alle bestanden naar een andere schijf

```
A>COPY TEST.BAS B:
```

kopiëe het bestand TEST.BAS naar een andere schijf

```
COPY TEST.BAS B:TEST1.BAS
```

kopiëer bestand TEST.BAS naar een andere schijf en geef het daar de naam TEST1.BAS

```
A>COPY TEST.TXT TEST1.TXT
```

kopiëer de inhoud van TEST.TXT naar een ander bestand met de naam TEST1.TXT

Deze kopiëer-opdrachten kwamen we in één of andere vorm al eerder tegen.

Zowel de eerste bestandsaanduiding (die aangeeft waar vandaan moet worden gekopiëerd) als de tweede bestandsaanduiding (die aangeeft waar naar moet worden gekopiëerd) mogen volledig als een bestandsaanduiding worden opgebouwd. Deze aanduidingen mogen dus vraagtekens en sterretjes bevatten. Zo kan men bijvoorbeeld op een gemakkelijke wijze een groep van bestanden kopiëren. Enkele voorbeelden:

`A>COPY *.TXT B:`	kopiëert alle bestanden met typering TXT naar een andere schijf.
`A>COPY TEST?.BAS B:`	kopiëert alle bestanden met typering BAS die een naam hebben van vijf letters waarvan de eerste vier gelijk zijn aan TEST, naar een andere schijf.
`A>COPY T*.* Q*.*`	alle bestanden met een naam, beginnend met een T, worden op dezelfde schijf gekopiëerd. In de nieuwe namen is deze eerste T ver-

vangen door een Q. Zo wordt een bestand TEST.BAS bijvoorbeeld gekopiëerd naar een bestand QEST.BAS.

A>COPY ?A?? ?B??

alle bestanden met een naam van vier letters waarvan de tweede letter een A is, worden gekopiëerd. De nieuwe bestanden krijgen de namen van oude bestanden met dit verschil dat A op de tweede plaats door een E is vervangen.

Bij deze kopiëerkommando's kan het gevaar schuilen dat een bestand wordt gekopiëerd onder een reeds in gebruik zijnde naam. Het bestand met deze naam wordt dan verwijderd ten behoeve van het nieuwe bestand. Dit is natuurlijk niet altijd de bedoeling...

Een kopiëerkommando als:

```
A>COPY *.* *.*
```

of

```
A>COPY TEST.* TEST.*
```

heeft tot gevolg dat bestanden op ZICHZELF worden gekopiëerd. De meeste MSX-DOS-systemen laten dit toe. Bij grote bestanden (groter dan ongeveer 60 kilobytes) kan dit echter problemen geven. Het is in dat geval mogelijk dat gedeelten van bestanden, soms zonder foutmelding, verloren gaan. Het kopiëren-opzichzelf van bestanden is dus sterk af te raden.

4 samenvoegend kopiëren

Het is mogelijk om meerdere bestanden tijdens het kopiëren samen te voegen tot één groot bestand. In dat geval worden de verschillende inhouden van de betreffende bestanden achter elkaar geplaatst en in één bestand overgenomen. Een voorbeeld:

```
A>COPY *.TXT ALLES.TXT
```

Deze kopiëeropdracht kopiëert de inhoud van alle bestanden met een typering TXT achter elkaar in één groot bestand, genaamd ALLES.TXT.

Het COPY-kommando staat het gebruik van een plus-teken toe om bestanden 'bij elkaar op te tellen'. Een voorbeeld:

```
A>COPY TEKST1.TXT+TEKST2.TXT TOTAAL.TXT
```

Deze opdracht kopiëert de inhoud van bestand TEST1.TXT, gevolgd door de inhoud van TEKST2.TXT in het bestand TOTAAL.TXT. Het kommando:

```
A>COPY A:TEKST1.TXT+B:TEKST2.TXT A:TOTA
AL.TXT
```

kombineert op eenzelfde wijze twee bestanden die echter van verschillende schijven afkomstig zijn. Tijdens uitvoering van dit laatste kommando zal de computer twee maal om een schijfwisseling vragen (tenzij u natuurlijk twee schijveneenheden aan de computer heeft gekoppeld).

Het samenvoegend kopiëren kent vele mogelijkheden. Zo heeft het kommando:

```
A>COPY *.TXT+*.LST *.PRN
```

tot gevolg dat van elk bestand met typering TXT een in naam overeenkomstig bestand met typering LST wordt opgezocht. De samenvoeging van deze twee teksten komt dan terecht in een gelijknamig bestand met typering PRN.

Wanneer bij het samenvoegend kopiëren geen bestemming wordt opgegeven aan het kopiëerkommando, wordt de eerstgenoemde file als bestemming genomen. Zo heeft het kommando:

```
A>COPY TEKST1.TXT+TEKST2.TXT
```

tot gevolg dat de samenvoeging van TEKST1.TXT en TEKST2.TXT in het bestand TEKST1.TXT terechtkomt. De originele TEKST1.TXT gaat bij deze kopiëeropdracht dus verloren.

Ook wanneer men niet samenvoegend kopiëert, neemt het kopiëerkommando bij gebrek aan bestemming altijd het eerste bestand. Een bevel als:

```
A>COPY TEST.TXT
```

heeft bijvoorbeeld tot gevolg dat het bestand op zichzelf wordt gekopiëerd. Praktisch merkt men daar meestal niets van.

Een kommando als:

```
A>COPY *.LST+*.PRN
```

heeft tot gevolg dat voor elk bestand met typering LST een bijbehorend bestand met typering PRN wordt gezocht. Beide bestanden worden samengevoegd en onder de typering LST (eerste naam) weer op schijf gezet. Alle originele LST bestanden gaan hierbij verloren.

5 speciale bestandsnamen

Het MSX-DOS kent enkele voorbestemde bestandsnamen. Deze namen kunt u zelf niet gebruiken om gegevens in op te slaan.

Deze bestandsnamen zijn:

CON	geeft de beeldscherm/toetsenbord-combinatie aan
AUX	geeft de communicatiepoort aan (RS-232-C indien aanwezig)
LST	geeft de printer aan
PRN	geeft eveneens de printer aan
NUL	geeft een loos afvoerkanaal aan

Deze bestandsnamen zijn niet door de gebruiker op schijf aan te leggen onder MSX-DOS, ook niet met één of andere typering. Deze bestandsnamen dienen slechts om de MSX-DOS-kommando's, in het bijzonder het COPY-kommando, extra mogelijkheden te geven.

CON

Deze aanduiding geeft de beeldscherm/toetsenbord-combinatie aan. Door ingave van het kommando:

```
A>COPY CON TEKST.TXT
```

kopiëert men de toetsenbord-ingave naar bestand TEKST.TXT. Na ingave van dit kommando kunnen regels tekst worden ingegeven. Wanneer dan uiteindelijk een CTRL-Z wordt ingegeven (tegelijk CTRL en Z indrukken), gevolgd door een RETURN, wordt de ingegeven tekst

in het bestand gezet en verschijnt de normale prompt weer. Door vervolgens in te geven:

```
A>COPY TEKST.TXT CON
```

wordt deze tekst naar het beeldscherm gekopiëerd; de tekst die zojuist werd ingegeven en die in bestand TEKST.TXT staat, verschijnt op beeld.

LST of **PRN**

Deze aanduiding geeft de printer aan. Door ingave van het kommando:

```
A>COPY TEKST.TXT LST
```

kopiëert men de inhoud van bestand TEKST.TXT naar de printer; de tekst die in een vorig voorbeeld werd ingegeven, wordt nu op de printer afgedrukt.

Door ingave van het kommando:

```
A>COPY CON PRN
```

kopiëert men de via het toetsenbord ingegeven gegevens naar de printer. Na ingave van een aantal regels tekst kan men de CTRL-Z ingeven, waarna de tekst onmiddellijk op de printer wordt afgedrukt.

AUX

Wanneer uw MSX computer een communicatie-uitgang (RS-232-C) heeft, kunt u deze aansturen met behulp van de aanduiding AUX. Wanneer u een tweede printer aan deze uitgang heeft gekoppeld, kunt u deze aktiveren door bijvoorbeeld:

`A>COPY TEKST.TXT AUX`	(de inhoud van TEKST.TXT wordt afgedrukt)
`A>COPY CON AUX`	(de ingave wordt na CTRL-Z op de printer afgedrukt).

Met deze uitgang kan men bijvoorbeeld ook een tweede computer aankoppelen. Met het COPY-kommando kunnen tekstuele gegevens dan zeer gemakkelijk worden verzonden van de ene naar de andere computer. Wanneer beide computers MSX computers zijn, kan een bestand met daarin tekstuele gegevens bijvoorbeeld als volgt worden overgezonden:

```
COMPUTER 1:    A>COPY TEKST.TXT AUX
COMPUTER 2:    A>COPY AUX TEKST.TXT
```

De gegevens, verzonden vanuit computer 1, komen uiteindelijk in bestand TEKST.TXT van computer 2 terecht.

Het overhevelen van andere dan tekstuele gegevens op deze wijze, alsmede het op de juiste wijze koppelen van twee voor elkaar vreemde computers is nog een hele kunst. Hierop kunnen we in dit bestek niet nader ingaan.

NUL

Het kan zijn dat een bevel onder MSX-DOS een bestandsaanduiding nodig heeft terwijl wij deze niet willen geven. In dat geval passen we de aanduiding NUL toe. Twee praktijkvoorbeelden:

Wanneer we een bestand op leesbaarheid willen controleren, dus wanneer we willen onderzoeken of een bestand nog wel helemaal vanaf schijf door de computer is in te lezen, kunnen we dit als volgt doen:

```
A>COPY TEST.BAS NUL
```

We kopiëren het bestand naar 'niets'. Hiervoor moet de computer echter wel het bestand TEST.BAS volledig inlezen. Wanneer we geen leesfout-melding krijgen, is bewezen dat het bestand door de computer nog te benaderen is.

Wanneer we een leeg bestand op schijf willen toewijzen, dan kan dat als volgt:

```
COPY NUL+NUL TEST.TXT
```

Een leeg bestand, genaamd TEST.TXT, wordt aangemaakt. De aanduiding NUL+NUL is meestal nodig omdat (een foutje van COPY) anders geen bestand wordt aangemaakt.

6 binair en ASCII (moeilijk)

Bestanden onder MSX-DOS kunnen een variëteit aan gegevens bevatten. Het kan zijn dat een bestand bijvoorbeeld alleen maar tekst bevat. Een bestand als bijvoorbeeld COMMAND.COM bevat echter bijvoorbeeld een kompleet machinetaal-programma.

Het kopiëren van bestanden met verschillende types van gegevens geeft over het algemeen geen problemen. Een uitzondering hierop vormt het kopiëren via de communicatie-uitgang (AUX), dat zonder verdere hulpmiddelen alleen met tekstbestanden mag geschieden.

Wanneer men echter bestanden aan elkaar gaat koppelen (samenvoegend kopiëren), kunnen problemen onstaan.

Wanneer men twee tekstbestanden samenvoegend wil kopiëren, kan dat bijvoorbeeld als volgt:

```
A>COPY TEKST1.TXT+TEKST2.TXT TEKST3.TXT
```

De bestanden TEKST1.TXT en TEKST2.TXT worden achter elkaar gezet en onder de naam TEKST3.TXT weer op schijf gezet.

Om te bepalen waar de tekstbestanden hun laatste karakter hebben staan, is er een einde-bestand-karakter (CTRL-Z) achter de tekst opgenomen. Dit einde-bestand-teken genereert men bij het kommando COPY CON TEST.TXT bijvoorbeeld door de CTRL- en de Z-toets tegelijk in te drukken.

Bij het normale samenvoegend kopiëren worden de samen te voegen bestanden afgezocht tot aan het eerste CTRL-Z-karakter. Alleen het gedeelte vanaf het begin tot aan het CTRL-Z-karakter wordt voor het kopiëren dan gebruikt.

Echter, een enkele maal kan de wat verder gevorderde amateur de noodzaak hebben om twee niet-tekstbestanden aan elkaar te koppelen. Dat kan bijvoorbeeld het geval zijn wanneer een groot machinetaalprogramma dient te worden samengesteld.

Met de /B-optie kan men in dat geval aangeven dat bestanden BINAIR (dus niet tekstueel) dienen te worden geïnterpreteerd. Het volgende voorbeeld kopiëert twee bestanden op binaire wijze naar één nieuw bestand:

```
A>COPY BEST1.COM/B+BEST2.COM BEST3.COM
```

De /B-optie blijft binnen een kopiëer-kommando geldig totdat een /A-optie wordt gegeven. Deze blijft op zijn beurt weer geldig binnen een kopiëerkommando totdat de /B-optie weer gevonden wordt.

Met de /A-optie (de A staat voor ASCII, een kodering voor een tekstuele opslagwijze) geeft men aan, dat een bestand een tekstuele inhoud heeft.

Wanneer men met de /B-optie kopiëert, zoekt de computer niet naar het eerste voorkomen van het einde-bestands-karakter (CTRL-Z) maar kopiëert het systeem gewoon het gehele bestand.

Van deze eigenschap kan men gebruik maken om bijvoorbeeld twee einde-bestands-karakters achter elkaar aan het einde van een bestand te plaatsen. Wanneer dit nodig is, kan men dat bereiken door ingave van bijvoorbeeld:

```
A>COPY TEKST.TXT/B TEKST.TXT/A
```

Het bestand TEKST.TXT wordt in eerste instantie binair ingelezen. Het einde-bestands-karakter, de CTRL-Z achter de inhoud van het bestand, wordt niet als zodanig beschouwd en dus gewoon mee ingelezen.
Het bestand TEKST.TXT wordt in tweede instantie, dit keer als tekstbestand, over het oude bestand heen terug geschreven. Omdat het nu als tekstbestand wordt beschouwd, wordt aan het einde van het bestand een CTRL-Z-karakter (einde-bestands-karakter) toegevoegd. In het nieuwe TEST.TXT staan per saldo nu dus twee CTRL-Z-tekens aan het einde.

9.2 De extrinsieke kommando's van MSX-DOS

Behalve intrinsieke kommando's kent het MSX-DOS ook extrinsieke kommando's. Deze kommando's horen niet bij het operating system, maar zijn daaraan toegevoegd als machinetaalprogramma's.

Extrinsieke kommando's zijn als zodanig te herkennen aan de typering COM (afkorting van COMMAND). Met het kommando

```
A>DIR *.COM
```

kunnen dus precies alle extrinsieke kommando's worden opgevraagd.

De extrinsieke kommando's bevatten machinetaalprogramma's. Tekstverwerkende programma's, databases en dergelijke programmatuur worden meestal in de vorm van COM-files aangeboden. Deze programma's zijn dan op te starten door alleen de naam van de COM-file (zonder de typering COM) in te geven.

Een COM-file vervangt bij het opstarten het controlerend programma van het operating system (COMMAND.COM) en start dit controlerende programma bij beëindiging van zijn funktie weer op (een WARM BOOT). In een vorig hoofdstuk startten we in feite al een COM-file op door ingave van:

```
A>COMMAND
```

We vervingen het controlerende programma van het operating system in dat geval echter door zichzelf waardoor er niets speciaals gebeurde.

Omdat extrinsieke kommando's in feite bestaan uit machinetaalprogramma's die niet vast bij het operating system horen, kunnen deze in het bestek van dit boek verder ook niet worden behandeld. Wanneer u een hoeveelheid extra programmatuur in de vorm van COM-files aanschaft, is deze meestal voorzien van een uitgebreide handleiding.

MSX-DOS biedt ons de mogelijkheid om verzamelingen van veel achter elkaar voorkomende kommando's samen te vatten in een nieuw, eenvoudig kommando. We kunnen onder MSX-DOS dus vrij eenvoudig eigen kommando's maken, gebruik makend van de onder MSX-DOS aanwezig zijnde kommando's.

10.1 Kommando's vereenvoudigen

We kunnen een schijf kopiëren door ingave van

```
A>COPY A:*.* B:
```

Wanneer men deze ingave echter veel te ingewikkeld vindt, kan men voor deze ingave een eenvoudiger kommando ontwerpen en wel als volgt:

```
A>COPY CON C.BAT
COPY A:*.* B:
^Z    (CTRL-Z tegelijk intoetsen)
A>
```

We hebben met behulp van het COPY-kommando een klein tekstbestandje ontworpen met daarin het kopiëerkommando. Dit tekstbestandje bevat het moeilijke kopiëerkommando dat we voortaan op een eenvoudige wijze willen kunnen oproepen en heeft de typering BAT (afkorting van BATCH) gekregen.

Voortaan kunnen we het kopiëren opstarten door ingave van alleen:

```
A>C
```

Het MSX-DOS zoekt bij deze ingave naar het BAT-bestand met de naam C (en natuurlijk de typering BAT) en voert vervolgens de kommando's uit die in dat bestandje staan.

We hebben nu voor het kopiëerkommando een nieuw, heel eenvoudig kommando ontworpen, namelijk het kommando C.

Wanneer we veelvuldig de inhoud van een schijf willen raadplegen op een verkorte wijze, dan kunnen we steeds ingeven:

```
A>DIR/W/P
```

waarna de inhoud wordt getoond. Gemakkelijker is het om éénmaal de BAT-file D.BAT te ontwerpen en wel als volgt:

```
A>COPY CON D.BAT
DIR/W/P
^Z
A>
```

Nu hoeven we voor een inhoudsopgave alleen nog maar de letter D, gevolgd door een RETURN, in te geven.

Een BAT-file mag ook meerdere kommando's bevatten. Een voorbeeld:

```
A>COPY CON K.BAT
PAUSE PLAATS DE ORIGINELE SCHIJF
DIR/W/P
PAUSE GEEF RETURN VOOR HET KOPIEREN
COPY A:*.* B:
PAUSE PLAATS DE ORIGINEEL SCHIJF
REM HET KOPIEREN IS TEN EINDE
BASIC
^Z
A>
```

Het bovenstaande voorbeeld geeft aan, hoe door middel van het vervaardigen van een BATCH-file een opvolging van kommando's kan worden vastgelegd. Het kommando K heeft voortaan tot gevolg dat het systeem netjes om de originele schijf vraagt, daar eerst een inhoud van laat zien, daarna pas op een teken van u gaat kopiëren, uiteindelijk weer om de originele schijf vraagt, meldt dat het kopiëren ten einde is en automatisch naar MSX-BASIC overschakelt.

Merk op dat de kommando's REM en PAUSE hier daadwerkelijk een funktie krijgen.

10.2 Variabelen in BAT-FILES

Men kan in een batch-bestand ook variabelen opnemen. Deze variabelen

zijn:

%0 geeft vast de naam van het kommando weer
%1 eerste na het kommando ingegeven variabele
%2 tweede na het kommando ingegeven variabele
.
.
.
%9 negende na het kommando ingegeven variabele.

Deze variabelen worden bij uitvoeren van het betreffende kommando vervangen door de na het BAT-kommando ingegeven gegevens. Een voorbeeld:

```
A>COPY CON TEST.BAT
REM KOMMANDO %0
DIR %1
PAUSE GEEF RETURN VOOR VERWIJDEREN
DEL %1
REM EINDE KOMMANDO %1
^Z
A>
```

In bovenstaand voorbeeld werd een kommando ontworpen dat eerst de te verwijderen bestanden laat zien en pas na een RETURN-toets de bestanden daadwerkelijk verwijdert. De bestandsaanduiding dient als eerste variabele aan dit kommando te worden opgegeven. Wil men bijvoorbeeld met behulp van dit nieuwe kommando alle bestanden met typering BAS verwijderen, dan geeft men in:

```
A>TEST *.BAS
```

Gedurende de uitvoering van het door ons samengestelde batch-kommando zullen de variabelen %0 en %1 worden vervangen door TEST (de naam van het kommando) en *.BAS (de na TEST ingegeven typering).

10.3 Automatisch opstarten

Wanneer men de computer vanuit MSX-DOS automatisch bij het opstarten een bepaalde taak wil laten uitvoeren, dan dient men deze taak

in een speciaal batch-bestand op te nemen. De naam van dit speciale batch-bestand is AUTOEXEC.BAT.

Wanneer men bijvoorbeeld direkt na het opstarten eerst de datum bij wil stellen en daarna het MSX-BASIC programma MENU.BAS wil opstarten, dan doet men dit als volgt:

```
A>COPY CON AUTOEXEC.BAT
DATE
BASIC MENU.BAS
^Z
A>
```

Het aldus ontworpen kommando AUTOEXEC wordt bij het opstarten van het systeem automatisch uitgevoerd. We kunnen dit testen docr de computer na vervaardiging van dit kommando eerst uit en dan weer aan te zetten.

Het is niet zinvol om in AUTOEXEC.BAT variabelen te specificeren.

10.4 Voorbeelden

Er zijn talloze voorbeelden te bedenken waarbij het ontwerpen van BAT-files goed van pas komt. In deze paragraaf volgen enkele, uitgewerkte voorbeelden van batch-bestanden.

B.BAT: automatisch opstarten BASIC

De volgende BAT-file heeft als resultaat dat het MSX-BASIC automatisch wordt opgestart wanneer onder MSX-DOS de letter B wordt ingegeven. Behalve een BAT-file moet ook een klein BASIC-programma worden geschreven.

Tik eerst onder MSX-DOS in:

```
A>COPY CON B.BAT
BASIC B.BAS
^Z              (druk de toetsen CTRL
A>               en Z tegelijk in)
```

Start hierna het MSX-BASIC op:

```
A>BASIC
```

De MSX-BASIC-melding verschijnt. Geef het volgende programma in:

```
NEW
Ok
10 KEY 1,"CALL SYSTEM"+CHR$(13)
20 NEW
SAVE "B.BAS"
Ok
CALL SYSTEM
A>              (het MSX-DOS is weer
                 geactiveerd)
```

Vanaf nu kan het basic worden opgestart met de ingave:

A>B

Funktietoets 1 zorgt in MSX-BASIC voor een eenvoudige terugkeer naar het MSX-DOS.

WIDTH.BAT: instellen beeldscherm

Onder MSX-BASIC kan de beeldschermbreedte met het kommando WIDTH XX worden ingesteld waarbij op de plaats van XX dan de gewenste breedte dient te worden opgegeven. Onder MSX-DOS dient deze breedte met het MODE XX kommando te worden ingegeven. Om verwarring te voorkomen, kan een WIDTH-kommando onder MSX-DOS worden samengesteld en wel als volgt:

Tik onder MSX-DOS in:

```
A>COPY CON WIDTH.BAT
MODE %1
^Z              (druk de CTRL en de
A>               Z-toets tegelijk in)
```

Vanaf nu kan het WIDTH-kommando ook onder MSX-DOS worden ingegeven, probeer maar eens met b.v.:

```
A>WIDTH 30
```

K.BAT: instellen MSX-DOS toetsen

Met de volgende BAT-file kan door alleen de ingave van de letter K de funktietoetsen op het MSX-DOS worden afgesteld. Behalve een BAT-file moet ook een BASIC-programma worden geschreven.

Tik eerst onder MSX-DOS in:

```
A>COPY CON K.BAT
BASIC K.BAS
^Z             (toets de CTRL en de
A>              Z-toets tegelijk in)
```

Activeer nu eerst het MSX-BASIC en schrijf het volgende programma:

```
A>BASIC  (de MSX-BASIC melding
NEW          verschijnt)
Ok
10 DATA 1,"CALL SYSTEM"
20 DATA 2,"DIR/P"
30 DATA 3,"COPY"
40 DATA 4,"FORMAT"
50 DATA 5,"BASIC"
60 DATA 6,"ERASE"
70 DATA 7,"RENAME"
80 DATA 8,"TYPE"
90 DATA 9,"MODE"
100 DATA 10,"DATE"
110 DATA -1
120 READ A
130 IF A=-1 THEN CALL SYSTEM
140 READ A$:KEY A,A$+CHR$(13)
```

```
150 GOTO 120
SAVE "K.BAS"
Ok
```

Uiteraard mogen in programmaregels 10-100 de funktietoetsen naar keuze worden ingedeeld.

HELP.BAT een geheugensteuntje

De volgende BAT-file geeft u een kort overzicht van de aanwezige MSX-kommando's als een geheugensteuntje. Alleen het woord HELP hoeft maar te worden ingeegeven.

Tik onder MSX-DOS in:

```
A>COPY CON HELP.BAT
TYPE HELP.TXT
^Z              (toets de CTRL- en de
A>               Z-toets tegelijk in)
```

Tik vervolgens onder MSX-DOS in:

```
A>COPY CON HELP.TXT
-------------------------------------
DIR /W /P   inhoudsopgave schijf
COPY        kopieren van bestanden
TYPE        uittikken tekstbestand
REN(AME)    bestanden anders noemen
DEL         verwijderen van bestanden
ERASE       als DEL
DATE        instellen van de datum
TIME        instellen van de tijd
PAUSE       pauze tot toetsingave
REM         opmerking
VERIFY      ON/OFF controle aan/uit
BASIC       naar BASIC toe
FORMAT      formatteren schijven
MODE        beeldschermbreedte
-------------------------------------
^Z          (geef de CTRL- en de Z-
A>           toets tegelijk in)
```

In het bovenstaande voorbeeld dienen ook de strepen te worden ingetoetst. Probeer de BAT-file nu uit door ingave van:

A>HELP

Het MSX-DOS kent enkele bijzonderheden die in dit hoofdstuk worden samengevat.

11.1 CTRL-P/CTRL-N

Wanneer men dit wenst, kan men met een eventueel aangesloten printer alle gegevens die op beeldscherm verschijnen, meteen laten afdrukken.

Deze print-optie kan men door ingave van CTRL-P (de CTRL- en de P-toets tegelijk indrukken) inschakelen. Op eenzelfde wijze schakelt men deze optie ook weer uit met CTRL-N.

PAS OP: nooit de CTRL-P optie gebruiken wanneer geen printer is aangesloten. MSX-DOS 'hangt zich op' in dat geval.

11.2 Besturingstoetsen onder MSX-DOS

Het is in MSX-DOS mogelijk om de ingave enigszins te besturen. Een per ongeluk fout ingetoetst kommando behoeft niet altijd te worden overgetikt maar kan eventueel worden aangepast.

De volgende besturingstoetsen zijn aanwezig:

pijl naar rechts	herhaalt teken voor teken het laatst ingegeven kommando.
pijl naar links	verwijderdt één karakter.
pijl naar boven	maakt de regel helemaal leeg.
pijl naar beneden	herhaalt het volledige laatste kommando.
DEL	verwijdert het volgende karakter van het laatst ingegeven kommando.
INS	voegt de volgende karakters tussen.
BS	verwijdert het laatst ingegeven karakter (als pijl naar links).
SELECT	na SELECT moet een karakter worden ingegeven. Het laatst ingegeven kommando wordt tot aan dit karakter overgenomen.
CLS	na CLS moet een karakter worden ingegeven. Het vorige kommando wordt tot aan dit karakter overgeslagen.

HOME maakt het tot nu toe ingegeven bevel tot het laatst ingegeven bevel.

Een voorbeeld: stel, we toetsen in:

```
A>COPY:*.BSA B:
```

Na uitvoering van dit kommando blijkt dat het fout was. We willen dit kommando nogmaals uitvoeren; BSA moet echter BAS worden.

Wanneer we de pijl naar beneden ingeven, zien we het gehele bevel weer tevoorschijn komen. Dit is echter niet de bedoeling; met de pijl omhoog verwijderen we dit laatste kommando weer. Met de pijl naar rechts kunnen we eventueel teken voor teken het oude kommando weer te voorschijn halen.

Met een SELECT-B ingave (eerst SELECT, dan de letter B) kunnen we het oude kommando echter in één klap tot aan de letter B overnemen. Met de pijl naar rechts herhalen we vervolgens de letter B. Door nu de letters AS in te geven, gevolgd door de pijl naar beneden, hebben we het uiteindelijke bevel nu goed op scherm staan. Door nu een RETURN in te geven, wordt het bevel daadwerkelijk uitgevoerd.

Wanneer we de bevelen TIME en DATE uitvoeren, kunnen we de huidige tijd en datum op een zelfde wijze aanpassen.

Alleen voor diegenen die onder MSX-DOS bijzonder veel ingave-werk moeten doen, heeft het zin om deze edit-mogelijkheden te bestuderen. Voor alle anderen staat het aanleren van de verschillende funkties waarschijnlijk niet in verhouding tot het nut dat men er ooit van heeft.

11.3 Scheidingskarakters

Tot nu toe hebben we in de MSX-DOS bevelen als scheidingskarakter steeds de spatie gebruikt. MSX-DOS staat echter meer scheidingskarakters toe.

De volgende scheidingskarakters mogen worden gebruikt:

TAB , (komma) ; (puntkomma) = (gelijkteken)

Dus een kopiëerkommando mag bijvoorbeeld ook als volgt worden ingetoetst:

```
A>COPY;A:*.*=B:
```

of

```
COPY A:*.*,B:
```

11.4 Foutmeldingen van MSX-DOS

Het MSX-DOS kent verschillende foutmeldingen. Deze meldingen worden niet gekodeerd in de vorm van een foutnummer maar in de vorm van een tekstje gegeven.

De volgende foutmeldingen kunnen door het MSX-DOS operating system worden gegeven:

Vanuit MSXDOS.SYS (vanuit het kernel)

Terminate batch file (Y/N)?

> Tijdens de uitvoering van een BAT-file heeft u een CTRL-STOP of een CTRL-C ingegeven. U kunt de uitvoering van het batchbestand verder laten gaan (Y) of de uitvoering stoppen (N).

Insert DOS disk in default drive
and strike any key when ready

> Na de uitvoering van een extrinsiek kommando probeerde MSX-DOS een WARM BOOT. COMMAND. COM werd echter niet aangetroffen; waarschijnlijk dient een andere schijf te worden geplaatst.

Bad FAT, drive...

> Op de aangegeven schijveneenheid is een foute File Allocation Table aangetroffen. In de praktijk betekent dat, dat een schijf is verminkt. De schijf zal opnieuw moeten worden geformatteerd alvorens deze weer kan worden gebruikt.

Write protect error writing drive...

> Bij een poging, gegevens op een schijf te schrijven, konstateerde

MSX-DOS dat de schijf in de aangegeven schijveneenheid write protected is. Doordat het write-protect stickertje is opgeplakt of doordat het write-protect schuifje is verschoven heeft men klaarblijkelijk te kennen gegeven dat de gegevens op deze schijf niet mogen worden overschreven.

Not ready error reading (writing) drive...

Tijdens een poging om gegevens te lezen/schrijven, konstateerde MSX-DOS dat de betreffende schijveneenheid hiervoor niet klaar was. Misschien is de spanning van de eenheid niet ingeschakeld of staat de vergrendeling nog los.

Disk error reading/writing drive...

Tijdens een lees/schrijfpoging werd een fout op schijf gekonstateerd. Misschien is de schijf nog niet geformatteerd, is hij versleten of is de eenheid vervuild of ontsteld.

Abort, Retry, Ignore?

Deze melding volgt altijd direkt op de drie voorgaande foutmeldingen. Met ingave van een R (retry) kan men de schijfaktie nog eens proberen, bijvoorbeeld nadat men de storing heeft opgeheven. Met I (ignore) kan men aangeven dat de fout verwaarloosd moet worden (niet aan te bevelen). Met A kan men tenslotte aangeven dat het gehele kommando dient te worden beëindigd.

Vanuit COMMAND.COM (vanuit het controlerende programma)

Insert disk with batch file
and strike any key when ready

Tijdens het uitvoeren van de kommando's uit een batch-file kon MSX-DOS de batch-file plotseling niet meer vinden. Vermoedelijk is de schijf ondertussen verwisseld. De schijf met het batchbestand dient te worden geplaatst waarna een toets moet worden ingegeven.

Strike a key when ready...

Deze melding kan op verschillende plaatsen voorkomen. De computer verwacht dan een toetsingave van u nadat een bepaalde

aktie is voltooid.

Invalid drive specification

Een verkeerde schijveneenheid-aanduiding werd aangegeven (bijvoorbeeld F:)

Program too big to fit in memory

Er werd getracht om een programma te laden dat te groot is om in het geheugen van de computer te passen.

Bad command or file name

Deze hele bekende melding verschijnt wanneer MSX-DOS niet weet wat het met een bepaalde ingave aan moet.

File not found

De gespecificeerde file werd niet gevonden

Are you sure (Y/N)?

Deze melding volgt als antwoord op een 'gevaarlijke' opdracht waarbij veel gegevens verloren kunnen gaan. Wanneer u zeker van uw zaak bent, geeft u een Y (YES) in, anders een N (NO).

Rename error

Tijdens het hernoemen van bestanden konstateerde REN(AME) een fout.

Invalid parameter

Een kommando werd voorzien van de verkeerde toevoegingen (bijvoorbeeld een VERIFY A:)

Insufficient disk space

Op schijf werd niet meer voldoende vrije ruimte gevonden om een kommando te voltooien.

File creation error

Tijdens het toewijzen van een bestand ontdekte MSX-DOS een fout.

File cannot be copied into itself

Er werd een bestand naar zichzelf gekopiëerd in een situatie waarin dat niet mogelijk is.

Content of destination lost before copy

Tijdens het kopiëren kwam een bestand zowel als te kopiëren bestand als naar te kopiëren bestand voor. Voordat het COPY-kommando het bestand kon inlezen, werd het al in de oorspronkelijke vorm aangetast. De gegevens uit dit bestand zijn verloren gegaan.

Write error

Een algemene schrijffout werd gekonstateerd.

Invalid date
Enter new date

Een verkeerde datum werd ingegeven. MSX-DOS vraagt alsnog om de goede datum.

11.5 Fouten in MSX-DOS

Het MSX-DOS versie 1.00 met COMMAND versie 1.01 vertoont nog enige fouten. Deze fouten zijn bijna nooit hinderlijk, maar kunnen soms weleens verwarring wekken.

Met name het COPY-kommando vertoont in vele uitvoeringen nogal storende fouten. Zo geeft het bijvoorbeeld altijd de melding '1file copied' ook wanneer er geen enkele file werd gekopiëerd. Ook de ingave COPY NUL TEST.TXT werkt bijvoorbeeld niet. Indien een te kopiëren bestand niet bestaat, wordt daar vaak geen melding van gemaakt.

Het aanmaken van BATCH-FILES voor MSX-DOS gaat over het algemeen met het COPY-kommando maar erg moeilijk.

Het volgende programma maakt u het aanmaken van BATCH-files wel heel gemakkelijk. Behalve BATCH-FILES kunt u ook andere tekstuele informatie op schijf bijhouden met EDITOR.

EDITOR is ook geschikt om programma's die met de A-optie zijn gesaved (ASCII-FILES), te laden en te verwerken. Zo kunnen snel, gemakkelijk en foutloos eventuele massale wijzigingen worden doorgevoerd of kunnen bepaalde woorden snel door het gehele programma worden opgezocht.

Omdat EDITOR geen funkties heeft om tekst bijvoorbeeld uit te lijnen, is dit programma voor echte tekstverwerking wat minder geschikt alhoewel er toch ook op dat gebied nog legio toepassingen voor EDITOR zijn te bedenken.

EDITOR, een algemeen, eenvoudig en zeer gemakkelijk te bedienen tekstverwerker, biedt u de volgende mogelijkheden:

- het opbouwen van tekstbestanden van maximaal 255 regels
- het werken met regellengten tot maximaal 255 tekens
- het veranderen van tekstdelen, handmatig, maar ook automatisch
- het verwijderen en tussenvoegen van regels tekst
- het verkrijgen van een overzicht over de volle 255 karakters tekstbreedte (ondanks dat het beeldscherm maar 40 tekens breed is)
- het opzoeken van woorden of zinsdelen

Voor velen zal EDITOR, nadat het éénmaal een keer is ingetikt, een dankbare hulp vormen bij uiteenlopende tekstverwerkende activiteiten.

Het intikken moet een keer gebeuren, daar is niets aan te doen. Omdat het een erg lang programma is, zullen verscheidene uurtjes tikwerk het gevolg zijn.

Omdat bij vorige uitgaven, ondanks dat de programmalijsten goed waren, nogal veel reacties loskwamen van lezers die een fout meenden te hebben ontdekt in het programma, verzoeken wij u vriendelijk om tij-

dens het intikken uiterst nauwkeurig te werk te gaan en alle ingetikte regels nog eens goed te (laten) controleren.

Het gebruik van EDITOR is bijzonder eenvoudig:

Nadat het programma is geladen/ingetikt, dient u het kommando RUN in te toetsen. EDITOR vraagt u dan onmiddellijk om een regelnummer.

Dit regelnummer kan gelijk zijn aan 1-255 of aan 999. Regelnummer 999 heeft een speciale betekenis zoals we verderop zullen zien.

Wanneer we een regelnummer ingeven, verschijnt dat nummer in beeld en kunnen we een tekst ingeven. Onthoud bij het tekst ingeven de volgende funkties van de pijltoetsen:

pijl naar links/rechts	= door de regel heen 'wandelen'
pijl naar boven	= het gedeelte, op en rechts van de cursor wordt één plaats verder naar rechts geplaatst zodat een ruimte voor een tussen te voegen teken ontstaat
pijl naar beneden	= het teken op de cursor wordt verwijderd.

Met de RETURN-TOETS beëindigen we een regel.

Wanneer een regel groter wordt dan de breedte van het beeldscherm, zien we dat de regel links uit beeld schuift. Door de pijl naar links te gebruiken, kan deze tekst weer worden teruggehaald.

De tekst die uit het beeldscherm schuift, blijft wel bewaard. Zo is het mogelijk om regels van 255 tekens te maken.

Wanneer we in plaats van een regelnummer alleen een RETURN-TOETS ingeven, dan neemt de computer automatisch een nummer aan dat een waarde 5 hoger ligt.

Wanneer we bijvoorbeeld al enige regels op beeldscherm hebben staan en er moet een regel worden tussengevoegd, dan kan dat door eenvoudigweg een tussenliggend regelnummer in te geven. Staat de betreffende regel op beeldscherm of zou die regel op beeldscherm staan wanneer hij bestond, dan worden de overige regels netjes een regel opzij geschoven en kan de gekozen regel worden tussengevoegd of veranderd. Geven we echter een regelnummer op dat nog niet op beeld aanwezig

was, dan projekteert EDITOR slechts het stuk tekst dat om deze regel heen staat, inclusief eventueel de regel zelf. Door dan een extra RETURN te geven, wordt ingave of verandering van de gekozen regel alsnog mogelijk gemaakt.

Wanneer er meer dan 21 regels zijn ingegeven, zal EDITOR trachten om steeds de meest relevante regels op het scherm te laten zien.

Wanneer u veel regels heeft ingetoetst die langer zijn dan 36 tekens, dan mist u op een gegeven moment een stuk overzicht, de rechterdelen van de regels worden niet geprojekteerd.

Door nu de toets DEL in te toetsen bij ingave regelnummer, schuift EDITOR de tekst 18 tekens naar links, zodat u 18 tekens verder in de regel kunt kijken. EDITOR doet dat voor alle op beeld geprojekteerde regels. Deze funktie kan meerdere keren achter elkaar worden gebruikt totdat uiteindelijk zelfs de langste regels met hun einde in beeld zijn

Wanneer u een regel hebt geselekteerd en u wilt deze regel nu in één keer verwijderen, dan kunt u dit doen door ingave van de HOME-toets of de DEL-toets.

Met de HOME-toets bereiken we weer het omgekeerde, de tekst schuift als het ware 18 posities naar rechts toe.

Door ingave van regelnummer 999 worden we gekonfronteerd met een keuze. We kunnen de huidige tekst op schijf (band) zetten door ingave van de letter S. Na deze ingave vraagt EDITOR om de bestandsnaam waarna het veiligstellen gebeurt.

Met de L kunnen we op een zelfde wijze een tekstfile weer inlezen. Merk hierbij op dat EDITOR, teneinde tussenvoegen zo gemakkelijk mogelijk te maken, zelf regelnummers toekent aan de binnengelezen regels. EDITOR verdeelt de tekst als het ware over de 255 mogelijke nummers zodat tussen alle regels meestal wel een invoegruimte is.

Met de letter N kunnen we de tekst in het computergeheugen verwijderen. Na gebruik van de N-funktie kunnen we een gehele nieuwe tekst opzetten.

Met de letter V kunnen we tekstdelen opzoeken en eventueel veranderen. Nadat de letter V is ingetoetst, vraagt EDITOR om het te veran-

deren deel. Hierna vraagt EDITOR in welke nieuwe tekst dit gedeelte moet worden veranderd (geef alleen RETURN wanneer u slechts wilt zoeken). Tijdens het veranderen projekteert EDITOR de regels waarin wordt veranderd.

Wanneer u in een MSX BASIC programma het één en ander massaal wilt veranderen, ga dan als volgt te werk.

1. Save het programma met de A-optie, bijvoorbeeld:
   SAVE"TEST",A
2. Aktiveer de editor en haal het programma binnen met de L-funktie
3. Voer de betreffende veranderingen door met de V-funktie (of misschien wilt u alleen maar wat opzoeken...)
4. Schrijf het programma weer onder dezelfde of een andere naam met de S-optie
5. Haal daarna onder MSX BASIC met een LOAD het programma weer binnen (b.v. LOAD "TEST")

EDITOR is in een TOP-DOWN-struktuur geschreven en voldoende gedokumenteerd. Voor de fervente amateur is EDITOR misschien een uitdaging. Na doorgronding van het programma kunnen misschien zelfs nog allerlei mooie funkties worden toegevoegd...

EDITOR mag, bijvoorbeeld ten behoeve van vrienden en kennissen, op non-commerciële basis worden verspreid door iedereen. Verhandeling op commerciële basis op publikatie door derden zonder voorafgaande schriftelijke toestemming van de uitgever is verboden en strafbaar.

Uiteindelijk volgt hier dan de programmalijst van EDITOR.

```
10 REM ***************************
20 REM *      TEKST-EDITOR       *
30 REM *      ------------       *
40 REM *  (C)1985 STARK TEXEL    *
50 REM ***************************
60 REM
70 REM ***************************
80 REM *    HOOFDPROGRAMMA       *
90 REM ***************************
100 REM
```

```
110 COLOR 15,4,4:SCREEN 0:WIDTH 40:CLS:K
EY OFF:CLEAR 17500:DIM A$(0),T%(0),B%(0)
:GOSUB 990'TEKST SCHOON
120 GOSUB 170'INGAVE REGELNUMMER
130 IF S=2 THEN GOSUB 270'PROJEKTEER
140 IF S=1 THEN GOSUB 390'EDIT REGEL
150 IF S=0 THEN GOSUB 490'LOAD/SAVE/SCHO
ON/VERANDER
160 GOTO 120
170 REM ***************************
180 REM *   REGELNUMMERINGAVE     *
190 REM ***************************
200 REM
210 I$="0023003REGEL":IO$="":NU=1:GOSUB
1150:IF II$="999" THEN S=0:RETURN
220 IF PG THEN PO=PO+18*PG+18*(PO>237 AN
D PG=1)-18*(PO<19 AND PG=-1):GOSUB 270:G
OTO 210
230 IF II$="" AND R>0 THEN S=1:RETURN'ED
ITEN
240 G1=0:G2=0:R=VAL(II$):IF R>255 OR R=0
 THEN 210 ELSE S=2:FOR I=0 TO 21:G1=(G1<
>0 OR B%(I)=0 OR B%(I)<R):G2=(G2<>0 OR B
%(I)=0 OR B%(I)>R):IF B%(I)=R OR G1*G2 T
HEN S=1:RETURN ELSE NEXT I:IF B%(0)=INST
R(B$,"1") AND R<B%(0) THEN S=1:RETURN
250 IF B%(21)=256-INSTR(C$,"1") AND R>B%
(21) THEN S=1
260 RETURN
270 REM ***************************
280 REM *      PROJEKTEREN        *
290 REM ***************************
300 REM
310 LOCATE 0,0,0:ERASE T%:DIM T%(44):G1=
22:RG=0:G=256-R:FOR I=1 TO 22:IF G=256 T
HEN 320 ELSE G=INSTR(G,C$,"1"):IF G=0 TH
```

```
EN 320 ELSE G1=22-I:T%(G1)=256-G:G=G+1:N
EXT I
320 G2=21:G=R+1:FOR I=1 TO 22:IF G=256 T
HEN 330 ELSE G=INSTR(G,B$,"1"):IF G=0 TH
EN 330 ELSE G2=21+I:T%(G2)=G:G=G+1:NEXT
I
330 IF G2<G1 THEN CLS:RETURN
340 IF G1>10 OR G2-G1<22 THEN G=G1:GOTO
360
350 IF G2<34 THEN G=G2-21:GOTO 360 ELSE
G=11
360 FOR G=G TO G+21:LOCATE 0,RG:IF T%(G)
 THEN PRINT RIGHT$(STR$(T%(G)+1000),3);"
:";MID$(A$(T%(G)),PO,36);
370 PRINT SPACE$(40);:IF T%(G)=R THEN RE
=RG
380 B%(RG)=T%(G):RG=RG+1:NEXT G:RETURN
390 REM **************************
400 REM * EDITEN VAN EEN REGEL   *
410 REM **************************
420 REM
430 IF A$(R)="" THEN A$(R)=" ":MID$(B$,R
,1)="1":MID$(C$,256-R,1)="1"
440 GOSUB 270'EERST PROJEKTEREN
450 IF A$(R)=" " THEN A$(R)="":MID$(B$,R
,1)="0":MID$(C$,256-R,1)="0"
460 IO$=A$(R):I$="00"+RIGHT$(STR$(RE+100
),2)+"255"+RIGHT$(STR$(R+1000),3):NU=0:G
OSUB 1150:A$(R)=II$:IF II$="" THEN MID$(
B$,R,1)="0":MID$(C$,256-R,1)="0":GOSUB 2
70 ELSE MID$(B$,R,1)="1":MID$(C$,256-R,1
)="1"
470 R=R+5:IF R>255 THEN R=5
480 RETURN
490 REM **************************
500 REM * LOAD/SAVE/SCHOON/VER. *
```

```
510 REM **************************
520 REM
530 I$="0023001(L)OAD/(S)AVE/(V)ERANDER/
(N)IEUW":IO$="":NU=0:GOSUB 1150:IF II$="
" THEN GOSUB 1040:RETURN
540 IO$="LlSsNnVv":KE=(INSTR(IO$,II$)+1)
\2:IF KE=0 THEN 530 ELSE ON KE GOSUB 550
,550,770,820:RETURN
550 REM **************************
560 REM *    LOAD/SAVE TEKST     *
570 REM **************************
580 REM
590 IO$=F$:GOSUB 1040:I$="0023012NAAM TE
KSTBESTAND":GOSUB 1150:IF II$="" THEN GO
SUB 1040:RETURN ELSE F$=II$
600 ON KE GOSUB 610,700:RETURN
610 REM **************************
620 REM *   LAAD TEKSTBESTAND    *
630 REM **************************
640 REM
650 GOSUB 990:OPEN II$ FOR INPUT AS 1:RC
=1
660 IF EOF(1) THEN 680 ELSE LINE INPUT #
1,A$(RC):IF A$(RC)="" THEN 660
670 RC=RC+1:IF RC=256 THEN GOSUB 1040:RE
TURN ELSE 660
680 RC=RC-1:IF RC=0 THEN GOSUB 1040:RETU
RN ELSE ST=255/RC:FOR RD=255 TO 1 STEP -
ST:RQ=INT(RD):SWAP A$(RQ),A$(RC):MID$(B$
,RQ,1)="1":MID$(C$,256-RQ,1)="1":RC=RC-1
:NEXT RD:R=128:CLS:GOSUB 270'PROJEKTEER
690 GOSUB 1040:RETURN
700 REM **************************
710 REM *   SAVE TEKSTBESTAND    *
720 REM **************************
730 REM
```

```
740 OPEN II$ FOR OUTPUT AS 1:FOR RC=1 TO
 255:IF A$(RC)<>"" THEN PRINT #1,A$(RC)
750 NEXT RC:GOSUB 1040:RETURN
760 NEXT RC:GOSUB 1040:RETURN
770 REM *************************
780 REM *   MAAK TEKST SCHOON   *
790 REM *************************
800 REM
810 GOSUB 990:LOCATE ,,0:CLS:RETURN
820 REM *************************
830 REM *    VERANDER/ZOEK OP   *
840 REM *************************
850 REM
860 GOSUB 1040:I$="0023016VERANDER":IO$=
"":GOSUB 1150:IF II$="" THEN GOSUB 1040:
RETURN ELSE VA$=II$:IO$=II$:GOSUB 1040:I
$="0023016IN":GOSUB 1150:IF II$="" THEN
GOSUB 690:RETURN ELSE NA$=II$:VR=1:RG=0:
CLS
870 IF VR=256 THEN GOSUB 1090:GOSUB 270:
GOSUB 1040:RETURN ELSE VR=INSTR(VR,B$,"1
"):IF VR=0 THEN GOSUB 1090:GOSUB 270:GOS
UB 1040:RETURN ELSE GOSUB 900:IF CH THEN
 LOCATE 0,RG:PRINT RIGHT$(STR$(VR+1000),
3);":";MID$(A$(VR),PO,36):RG=RG+1
880 IF RG=22 THEN GOSUB 1090
890 VR=VR+1:GOTO 870
900 REM *************************
910 REM *  VERANDER EEN REGEL    *
920 REM *************************
930 REM
940 VP=1:CH=0
950 IF VP=256 THEN RETURN ELSE VP=INSTR(
VP,A$(VR),VA$):IF VP=0 THEN RETURN
960 CH=1:IF VA$=NA$ THEN RETURN ELSE V1$
=LEFT$(A$(VR),VP-1):V2$=NA$:V3$=MID$(A$(
```

```
VR),VP+LEN(VA$)-1):V3$=MID$(V3$,2)
970 IF LEN(V1$)+LEN(V2$)+LEN(V3$)>255 TH
EN IF LEN(V3$) THEN V3$=LEFT$(V3$,LEN(V3
$)-1): GOTO 970 ELSE V2$=LEFT(V2$,LEN(V2
$)-1):GOTO 970
980 A$(VR)=V1$+V2$+V3$:VP=VP+LEN(NA$):GO
TO 950
990 REM *************************
1000 REM *   SCHOONMAKEN TEKST    *
1010 REM *************************
1020 REM
1030 ERASE A$,B%:DIM A$(255):B$=STRING$(
255,"0"):C$=B$:PO=1:R=5:DIM B%(21):RETUR
N
1040 REM *************************
1050 REM * ONDERSTE REGEL SCHOON *
1060 REM *************************
1070 REM
1080 CLOSE:LOCATE 0,23:PRINT SPACE$(39);
:RETURN
1090 REM *************************
1100 REM *  RETURN-TOETS INGEVEN *
1110 REM *************************
1120 REM
1130 LOCATE 0,23:PRINT"(RETURN)";
1140 IF INKEY$="" THEN 1140 ELSE RG=0:CL
S:RETURN
1150 REM *************************
1160 REM *     INGAVEROUTINE      *
1170 REM *************************
1180 REM
1190 XX=VAL(LEFT$(I$,2)):YY=VAL(MID$(I$,
3,2)):TT=VAL(MID$(I$,5,3)):II$=IO$:IP=LE
N(II$):IP=IP-(IP=0):II$=II$+SPACE$(TT-LE
N(II$)):PG=0
1200 LOCATE XX,YY: PRINT MID$(I$,8);":";
```

```
LEFT$(II$,46-LEN(I$)-XX);
1210 X1=XX+LEN(I$)+IP-7:IF X1>38 THEN LO
CATE XX+LEN(I$)-6,YY:PRINT MID$(II$,X1-3
8,40-POS(0)):X1=39
1220 LOCATE X1,YY,1
1230 KK$=INKEY$:IF KK$="" THEN 1230 ELSE
 KK=ASC(KK$)
1240 IF KK=11 THEN PG=-1:II$="":RETURN E
LSE IF KK=127 THEN PG=1:II$="":RETURN
1250 IF KK=28 THEN IP=IP+1+(IP=LEN(II$))
:GOTO 1210
1260 IF KK=29 THEN IP=IP-1-(IP=1):GOTO 1
210
1270 IF KK=30 THEN II$=LEFT$(II$,IP-1)+"
 "+MID$(II$,IP,LEN(II$)-IP):PRINT MID$(I
I$,IP,40-X1);:GOTO 1210
1280 IF KK=31 THEN IF IP<TT THEN II$=LEF
T$(II$,IP-1)+MID$(II$,IP+1)+" ":PRINT MI
D$(II$,IP,40-X1);:GOTO 1210 ELSE 1210
1290 IF KK<>13 THEN 1340
1300 QQ=128
1310 IF QQ>TT THEN QQ=QQ/2:GOTO 1310
1320 FOR II=LEN(II$) TO 1 STEP -QQ:IF RI
GHT$(II$,QQ)=SPACE$(QQ) THEN II$=LEFT$(I
I$,LEN(II$)-QQ):NEXT II
1330 IF QQ=1 THEN RETURN ELSE QQ=QQ/2:GO
TO 1320
1340 IF KK<32+16*NU OR KK>127-70*NU THEN
 BEEP:GOTO 1210
1350 MID$(II$,IP,1)=KK$:PRINT KK$;:IP=IP
+1:IF IP>TT THEN IP=TT
1360 GOTO 1210
```

Sommige MSX-computers vertonen een wat ander geheugengebruik. In dat geval kan het met CLEAR gereserveerde string-geheugen op re-

gel 110 soms een probleem geven. Wanneer u een 'Out of memory' fout krijgt, verlaagt u dan het getal achter CLEAR op regel 110 zodanig dat deze fout niet meer optreedt. Een CLEAR 15000 is over het algemeen veilig.

Het verlagen van het getal achter CLEAR betekent echter wel het verlagen van het beschikbare opslaggeheugen voor het programma.

Wanneer u echt ALLES in bestanden wilt veranderen, heeft u een speciaal programma nodig. Wilt u bijvoorbeeld bestuderen hoe een BASIC-FILE is opgebouwd, hoe COMMAND.COM in elkaar zit of wilt u veranderingen in een bestand aanbrengen, dan kan dat niet zomaar. Een heel speciaal toegangsprogramma moet u hiertoe de mogelijkheid geven.

BYTES is zo'n programma. BYTE voor BYTE kunt u van alle bestanden de inhoud bestuderen, hexadecimaal maar ook tekstueel, en kunt u, wanneer u dit wenst, zelfs wijzigingen op byte-niveau toepassen.

Met name is het vaak grappig om te trachten, bepaalde meldingen die in COMMAND.COM en MSXDOS.SYS liggen opgeslagen, te veranderen naar de nederlandse taal. Met BYTES gaat dat erg leuk en de resultaten zijn verrassend.

Ook kan bytes behulpzaam zijn bij het tijdelijk onbruikbaar maken van bepaalde programma's. Door één of meer katastrofale wijzigingen in een programma aan te brengen, voorkomt u dat in een onbewaakt moment (op een hobby-beurs of iets dergelijks) bruikbare software van u wordt gekopiëerd. Uiteraard dient u de wijzigingen te kennen en te kunnen terugdraaien.

Zo kan een MSX-BASIC programma onbruikbaar worden gemaakt door het eerste byte, dat organiek op FF hexadecimaal (255 decimaal) staat, op een andere waarde te stellen.

BYTES verlangt van u dat u uiterst voorzichtig te werk gaat (experimenteer alleen op speciaal daartoe door u vervaardigde kopiëen) en dat u het hexadecimale talstelsel enigszins beheerst.

Het gebruik van EDITOR is erg eenvoudig:

Er zijn drie kommando's, te weten:
- het G-kommando voor het ophalen van de betreffende file
- het P-kommando om de file weer naar schijf te schrijven
- het D-kommando om gegevens te kunnen projekteren.

Zowel achter het P als achter het G-kommando moet u direkt de filenaam vermelden. Wilt u nadat u BYTES met RUN hebt opgestart, het

bestand COMMAND.COM laden, dan geeft u bijvoorbeeld in:

`GCOMMAND.COM` (achter elkaar)

Na de RETURN wordt de file ingelezen.

Wilt u later, na veranderingen, COMMAND.COM weer op schijf schrijven, dan geeft u in:

`PCOMMAND.COM` (achter elkaar)

Het bestand wordt dan weer teruggeschreven. Uiteraard kunt u het veranderde bestand ook onder een andere naam vastleggen.

Met het D-kommando kunt u zowel hexadecimaal als tekstueel de inhoud van een geladen file bestuderen. Geef hiertoe het D-kommando in, gevolgd door eventueel het (hexadecimale) adres waar vanaf moet worden gelijst. Het enkele D-kommando lijst verder vanaf het laatst aktieve adres. Het D-kommando kan op elk moment worden gestopt door een willekeurige toetsaanslag.

Door alleen een hexadecimaal adres in te geven, bewerkstelligt men dat de inhoud van dat betreffende adres wordt afgedrukt waarna een nieuwe inhoud kan worden opgegeven. Altijd geldt dat zowel een hexadecimale konstante als een ASCII-karakter kan worden ingegeven. Een karakter is (uiteraard) nooit langer dan één teken terwijl een hexadecimale konstante verplicht PRECIES twee cijfers moet bevatten. Eventueel dient de hexadecimale konstante van een voorloop-nul te worden voorzien.

Door alleen een RETURN-toets in te geven, blijft de inhoud onveranderd. Door nu weer de RETURN-toets in te geven, wordt het volgende adres behandeld zodat de file stap voor stap kan worden doorlopen en veranderd.

Na het aanbrengen van een verandering op een lokatie volgt altijd onmiddellijk de mogelijkheid om een volgende ingave, één byte verder in het bestand, te doen. Zo kunnen snel vertalingen worden doorgevoerd.

BYTES is in een TOP-DOWN-struktuur geschreven. De kenner herkent bij bestudering in een enkel geval zelfs een recursieve techniek.
De moeilijkheidsgraad van BYTES ligt niet zo hoog en elke wat verder

gevorderde amateur moet dit programma dan ook wel vrij snel kunnen doorgronden.

BYTES mag, bijvoorbeeld ten behoeve van vrienden of kennissen, op non-commerciële basis worden verspreid door iedereen. Verhandeling op commerciële basis of publikatie door derden zonder voorafgaande, schriftelijke toestemming van de uitgever is verboden en strafbaar.

Uiteindelijk volgt hier de programmalijst van BYTES.

```
10 REM ************************
20 REM *        BYTES         *
30 REM *        -----         *
40 REM * (C) 1985 STARK TEXEL *
50 REM ************************
60 REM
70 REM ************************
80 REM *    HOOFDPROGRAMMA    *
90 REM ************************
100 REM
110 CLEAR 20000:DIM Q$(0):CLS:WIDTH 40:C
$="GgPpDd":H$="0123456789ABCDEF012345678
9abcdef"
120 LINE INPUT K$:IF K$="" THEN K$=" "
130 B$=LEFT$(K$,1):B=INSTR(C$,B$):IF B=0
 THEN LOCATE 0,CSRLIN-1:GOSUB 490:GOTO 1
20
140 B=(B-1)/2+1:ON B GOSUB 200,300,380
150 GOTO 120
160 REM ************************
170 REM *   OPHALEN BESTAND    *
180 REM ************************
190 REM
200 F$=MID$(K$,2):IF F$="" THEN BEEP:RET
URN
210 OPEN F$ FOR INPUT AS 1:CLOSE:OPEN F$
 AS 1
```

```
220 FIELD 1,128 AS A$, 128 AS B$
230 R=INT(LOF(1)/256+.999):IF R=0 THEN C
LOSE:RETURN
240 ERASE Q$:DIM Q$(R*2)
250 FOR I=1 TO R:GET #1,I:Q$(2*I-1)=A$:Q
$(2*I)=B$:NEXT I:CLOSE:RETURN
260 REM **************************
270 REM *   SCHRIJVEN BESTAND    *
280 REM **************************
290 REM
300 F$=MID$(K$,2):IF F$="" THEN BEEP:RET
URN
310 OPEN F$ AS 1
320 FIELD 1,128 AS A$, 128 AS B$
330 FOR I=1 TO R:LSET A$=Q$(2*I-1):LSET
B$=Q$(2*I):PUT #1,I:NEXT I:CLOSE:RETURN
340 REM **************************
350 REM *    DISPLAY GEGEVENS    *
360 REM **************************
370 REM
380 IF R=0 THEN RETURN ELSE Z$=""
390 F$=MID$(K$,2):GOSUB 700:IF F=-1 THEN
 RETURN ELSE IF F$>"" THEN D=F
400 FOR J=1 TO 16:Q=D:GOSUB630:PRINT RIG
HT$(U$,4);":";:FOR I=1 TO 8:B1=D\128:B2=
D-128*B1:B1=B1+1:B2=B2+1
410 IF B1>2*R THEN D=0:PRINT:RETURN
420 K=ASC(MID$(Q$(B1),B2)):IF K>31 AND K
<128 THEN Z$=Z$+CHR$(K) ELSE Z$=Z$+"."
430 Q=K:GOSUB 630:PRINT RIGHT$(U$,2); "
";:D=D+1:IF INKEY$>"" THEN PRINT: RETURN
 ELSE NEXT I:PRINT " ";Z$:Z$="":NEXT J:R
ETURN
440 B=B-8
450 REM **************************
460 REM *   VERANDER GEGEVENS    *
470 REM **************************
```

```
480 REM
490 IF R=0 THEN RETURN ELSE IF K$=" " TH
EN K$=""
500 F$=K$:GOSUB 700:IF F=-1 THEN BEEP:RE
TURN ELSE IF F$>"" THEN D=F ELSE D=D+1
510 B1=D\128:B2=D-128*B1:B1=B1+1:B2=B2+1
520 IF B1>2*R THEN RETURN
530 Q=D:GOSUB 630:PRINT RIGHT$(U$,4);":"
;
540 K=ASC(MID$(Q$(B1),B2)):IF K>31 AND K
<128 THEN Z$=CHR$(K) ELSE Z$="."
550 Q=K:GOSUB 630:PRINT RIGHT$(U$,2);" "
;Z$;":";
560 LINE INPUT F$:IF F$="" THEN RETURN
570 IF LEN(F$)=1 THEN K=ASC(F$) ELSE GOS
UB 700:K=F:IF K>255 OR K<0 THEN BEEP:GOT
O 510
580 MID$(Q$(B1),B2,1)=CHR$(K):D=D+1:GOTO
 510
590 REM **************************
600 REM *     Q HEX NAAR U$      *
610 REM **************************
620 REM
630 U$="0000000"
640 IF Q>15 THEN Q=Q/16:GOSUB 640
650 U$=U$+MID$(H$,Q+1,1):Q=(Q-INT(Q))*16
:RETURN
660 REM **************************
670 REM *     F$ DEC NAAR F      *
680 REM **************************
690 REM
700 F=0:IF F$="" THEN RETURN ELSE FOR K=
1 TO LEN(F$):W=INSTR(H$,MID$(F$,K,1)):IF
 W=0 THEN F=-1:RETURN ELSE IF W>16 THEN
W=W-16
710 F=F*16+W-1:NEXT K:RETURN
```

Sommige MSX computers vertonen een wat ander geheugengebruik. In dat geval kan het met CLEAR gereserveerde string-geheugen op regel 110 soms een probleem geven. Wanneer u een 'Out of memory' fout krijgt, verlaagt u dan het getal achter CLEAR op regel 110 zodanig dat deze fout niet meer optreedt. Een CLEAR 18000 is over het algemeen veilig.

Het verlagen van het getal achter CLEAR betekent echter wel het verlagen van het beschikbare opslaggeheugen voor het programma.

Wanneer men meerdere programma's op een schijf heeft staan, dan wil men de mogelijkheid om en van deze programma's te kiezen vaak vereenvoudigen. Hiertoe maakt men meestal gebruik van een zogenaamd programmamenu. Een programmamenu is een programma dat u in staat stelt om andere programma's op te starten, meestal door slechts een nummer in te geven.

Het volgende programma is zo'n menuprogramma. De programma's die in het menu dienen te worden opgenomen, staan in DATA-statements achterin het programma. Van elk programma dient in een DATA-statement steeds de programmaomschrijving, gevolgd door de naam van het programma te worden opgenomen. Het programma EDITOR.BAS is bijvoorbeeld als volgt opgenomen:

```
350 DATA "EDITORPROGRAMMA","EDITOR.BAS"
```

Wanneer de naam van het porgramma niet wordt gegeven (" ") dan wordt de voorafgaande tekst als verduidelijkende tekst opgenomen.

Als eerste DATA-regel dient de naam van de schijf te worden opgenomen.

In het onderstaande programma zijn enkele voorbeeld-programma's opgenomen. Het staat u natuurlijk vrij om deze DATA-regels naar uw wensen aan te passen.

Wanneer u dit menu-programma bij het aanschakelen van het systeem automatisch wilt laten opstarten, maak dan als volgt een BATCH-file onder MSX-DOS aan:

```
COPY CON AUTOEXEC.BAT
BASIC MENU.BAS"
^Z
A>
```

(de toetsen CTRL en Z tegelijk indrukken)

Het programma moet in dat geval natuurlijk wel onder de naam MENU.BAS worden geSAVEd.

MENU mag, bijvoorbeeld ten behoeve van vrienden of kennissen, op non-commerciële basis worden verspreid door iedereen. Verhandeling

op commerciële basis of publicatie door derden zonder voorafgaande schriftelijke toestemming van de uitgever is verboden en strafbaar.

Uiteindelijk volgt hier de programmalijst van MENU.

```
10 REM ************************
20 REM *         MENU         *
30 REM *         ----         *
40 REM *  (C)1985 STARK TEXEL  *
50 REM ************************
60 REM
70 REM ************************
80 REM *    HOOFDPROGRAMMA    *
90 REM ************************
100 REM
110  RESTORE:CLEAR 200:WIDTH 40:SCREEN 0
:CLS:KEY OFF:DIM P$(20)
120 GOSUB 150'KEUZE AFDRUKKEN
130 GOSUB 220'KEUZE MAKEN
140 RUN P$'GEKOZEN PROGRAMMA HALEN
150 REM ************************
160 REM *    KEUZE AFDRUKKEN    *
170 REM ************************
180 REM
190 LOCATE 0,0:PRINT "KEUZEPROGRAMMA SCH
IJF:";:READ P$:PRINT P$
200 FOR I=3 TO 20:LOCATE 8,I:READ P$,P$(
I-2):IF P$(I-2)>"" THEN PRINT RIGHT$(STR
$(I+98),2);" ... ";P$ ELSE IF P$>"" THEN
 PRINT P$
210 NEXT I:RETURN
220 REM ************************
230 REM *  KEUZE KEUZE BEPALEN  *
240 REM ************************
250 REM
260 LOCATE 0,22:INPUT "UW KEUZE ";K
```

```
270 IF K<1 OR K>20 OR K>INT(K) THEN 280
ELSE IF P$(K)="" THEN 280 ELSE P$=P$(K):
RETURN
280 BEEP:LOCATE 0,22:PRINT SPACE$(39);:G
OTO 260 ELSE P$=P$(K):RETURN
290 REM ************************
300 REM * TE KIEZEN PROGRAMMA'S *
310 REM ************************
320 REM
330 DATA "TESTDISKETTE"
340 DATA "PROGRAMMA'S ALGEMEEN:","","",""
"
350 DATA "EDITORPROGRAMMA","EDITOR.BAS"
360 DATA "BYTESPROGRAMMA","BYTES.BAS"
370 DATA "TEKENPROGRAMMA","DRAWS.BAS"
380 DATA "","","ADMINISTRATIEVE PROGRAMM
A'S:","","",""
390 DATA "BOEKHOUDPROGRAMMA","BOEK.BAS"
400 DATA "TYPE-OEFENPROGRAMMA","TYPE.BAS
"
410 DATA "","","SPELLETJES:","","",""
420 DATA "BOTER, KAAS EN EIEREN","SPEL01
.BAS"
430 DATA "KLAVERJASSEN","SPEL02"
440 DATA "GALGJE","SPEL03"
450 REM ************************
460 REM *   EINDE DATA-GEBIED   *
470 REM ************************
480 REM
490 DATA ,,,,,,,,,,,,,,,,,,,,,
500 DATA ,,,,,,,,,,,,,,,,,,,,,
```

Aantekeningen

Aantekeningen

Aantekeningen

Aantekeningen

Aantekeningen

REAKTIES

Fouten en onduidelijkheden

Wanneer u in dit boek fouten of onduidelijkheden bent tegengekomen stellen wij het bijzonder op prijs dat u die aan ons doorgeeft. Daar kunnen wij met volgende drukken en andere uitgaven rekening mee houden.

Boeken en programma's

Schrijft u een boek of hebt u een goed programma gemaakt? Dan kunnen wij u misschien interessante auteurskondities aanbieden. Stuur nog geen materiaal, maar neem eerst even kontakt met ons op voor overleg.
Ons adres vindt u voorin dit boek.

Nieuwe onderwerpen

Wilt u een bepaald onderwerp graag behandeld zien in een uitgave? Schrijf het ons. Bij voldoende belangstelling kunnen wij daar dan rekening mee houden.

Graag horen we nader van u.

ENKELE MSX-uitgaven

serie: **UW MSX-COMPUTER DE BAAS**

MSX BASIC HANDBOEK voor iedereen, door A.C.J. Groeneveld
Een kompleet nederlandstalig handboek voor iedere MSX computergebruiker ISBN 90 6398 100 7

MSX ZAKBOEKJE door Wessel Akkermans
Een vlot geschreven naslagwerk na of naast het handboek. U vindt er o.a. in: niet computergerichte tabellen; de MSX-BASIC instructieset; diverse tabellen die het BASIC-programmeren kunnen versnellen; de Z80 instructieset; hardware-gegevens (connectoren) en een aantal programmaatjes ISBN 90 6398 888 5

MSX DISK HANDBOEK door A.C.J. Groeneveld
Handboek voor diskdrivebezitters om naast het grote handboek te gebruiken. Een zeer volledige behandeling van het disk-gebeuren zelf en de specifieke disk kommando's, uitgebreid met voorbeelden, tabellen en overzichten. Het handboek is aangevuld met interessante programma's, waaronder een tekentafelprogramma en een basisprogramma voor basisonderhoud ISBN 90 6398 407 3

MSX PRAKTIJKPROGRAMMA'S door Wessel Akkermans
Praktische programma's met waar nodig eerst een stukje theorie. Erg handig bij het maken van uw programma's. Een greep uit de onderwerpen: priemgetallen; zoeken en sorteren; trefwoordenlijsten; converteren van getallen; enz. ISBN 90 6398 437 5

MSX QUICK DISK handboek voor iedereen, door A.C.J. Groeneveld
Hèt handboek voor iedere QUICK DISK gebruiker. Uitvoerige behandeling van de sleutelwoorden aangevuld met duidelijke voorbeelden met listing ISBN 90 6398 254 2

SOFTWARE PLUS IN MSX

INTROTAPE MSX door A.C.J. Groeneveld.
Heeft u nog maar net een MSX computer gekocht en wilt u graag weten wat de computer kan en hoe u hem kunt leren programmeren? Deze cassette introduceert MSX op een uiterst vriendelijke en onderwijzende manier. U krijgt instructies hoe u de computer aan moet sluiten en de tape laden. Daarna volgt een demonstratie van de mogelijkheden in MSX, zoals het tekenen van sprites en het werken met de driestemmige toongenerator. Het geheel wordt afgesloten met twee 'les' gedeeltes. In anderhalf à drie uur weet u wat de MSX computer is, wat hij kan, en heeft u haast ongemerkt al wat regels geprogrammeerd.
ISBN 90 6398 148 1

MSX SCRIPT door Ton Weijters
Een menugestuurde nederlandstalige tekstverwerker. Het programma is geschikt om efficiënt grotere of kleinere teksten te bewerken. Pagina-indeling (regellengte, paginalengte, marge, inspringen, centreren, enz.) wordt door het programma verzorgd. Dit geldt ook voor de paginatelling, toptitel en het eventueel uitvullen van de regels. Ook corrigeren, zoeken, string-substitutie, blokken tekst verplaatsen, copiëren of verwijderen, onderstrepen en vet zetten, is mogelijk met dit programma.
ISBN 90 6398 189 9

MSX DRAWS door A.C.J. Groeneveld
Een tekenprogramma in MSX basic, waarmee u al binnen 10 minuten uw eerste tekening kunt maken. Draws werkt erg vriendelijk en maakt gebruik van alle grafische mogelijkheden van de MSX computer. U kunt met Draws zowel technisch als creatieve tekeningen maken. Het programma heeft een effectief bereik van ruim 30.000 bij 30.000 puntjes met mogelijkheden als lijnen, cirkels, krommen, inkleuren, vergroten, verkleinen, verschuiven, verdraaien en andere tekeningen invoegen
ISBN 90 6398 754 4

MSX DOS

handboek voor iedereen

Met het MSX-DOS dat door de systeemsoftware-expert Microsoft is ontwikkeld, is er eindelijk een goed besturingssysteem gekomen voor hobbycomputers. De professionele afkomst van MSX-DOS (het alom bekende MS-DOS) en de zeer grote mate van uitwisselbaarheid van programmatuur met het wereldberoemde CP/M operating system van DIGITAL RESEARCH, maken het MSX-DOS tot een instrument dat de MSX-computer met schijveneenheid ongekende mogelijkheden biedt. Zo kan de meeste professionele CP/M-software na vaak maar een kleine aanpassing ook onder het MSX-DOS operating system draaien. Software fabrikanten van bijvoorbeeld Pascal-compilers, Databases, tekstverwerkers en andere vrij zware programmatuur, kunnen heel eenvoudig hun software ook op de MSX-markt afstemmen. Zo kunnen zelfs MSX-computers met één of twee schijveneenheden en eventueel een kleine printer al snel als semi-professioneel systeem worden ingezet.
Met het MSX-DOS is de MSX-standaard een feit. Eindelijk een kompleet, professioneel totaalkonsept voor een breed scala van computers, variërend van kleine hobby-computers tot en met semi-professionele systemen. De bezitter van een MSX-computer, van welk merk dan ook, zit wanneer hij wat uit wil breiden niet meer op dood spoor. En dat was vóór de MSX-standaard maar al te vaak wèl het geval.
Het MSX-DOS-handboek geeft op de heldere wijze die u van de MSX handboekenreeks van Stark-Texel gewend bent, een totaalbeeld van de mogelijkheden van het MSX-DOS. Niet alleen het MSX-DOS wordt behandeld, maar voorafgaand hieraan wordt ook een inleiding op het begrip 'operating system' gegeven. Ook het MSX-DOS-handboek is dus een HANDBOEK VOOR IEDEREEN.

ISBN 90 6398 674 2